

N° d'enregistrement
au C.N.R.S.

THESE DE DOCTEUR-INGENIEUR :
Mathématiques

présentée
A L'UNIVERSITE PARIS VI

par
M. BOSSUET Gérard

pour obtenir
1e GRADE DE DOCTEUR INGENIEUR

SUJET de 1a THESE

ETUDE ET REALISATION D'OPERATEURS PERMETTANT PAR LEUR
ASSEMBLAGE LA CONSTRUCTION DE STRUCTURES MICROPROGRAMMEES GE-
NERALES OU SPECIALISEES. ETUDE PRATIQUE.

Soutenue le :

26

JUIN 1971

devant la Commission d'Examen

MM. R. SIESTRUNCK

J. VIGNES

J. LOMBARD

J. SUCHARD

M. COURTILLOT

Président

Examineurs

N° d'enregistrement
au C.N.R.S.

THESE DE DOCTEUR-INGENIEUR :
Mathématiques

présentée
A L'UNIVERSITE PARIS VI

par
M. BOSSUET Gérard

pour obtenir
1e GRADE DE DOCTEUR INGENIEUR

SUJET de 1a THESE

ETUDE ET REALISATION D'OPERATEURS PERMETTANT PAR LEUR
ASSEMBLAGE LA CONSTRUCTION DE STRUCTURES MICROPROGRAMMEES GE-
NERALES OU SPECIALISEES. ETUDE PRATIQUE.

Soutenue le : JUIN 1971

devant la Commission d'Examen

MM. R. SIESTRUNCK

J. VIGNES

J. LOMBARD

J. SUCHARD

M. COURTILLOT

Président
Examineurs



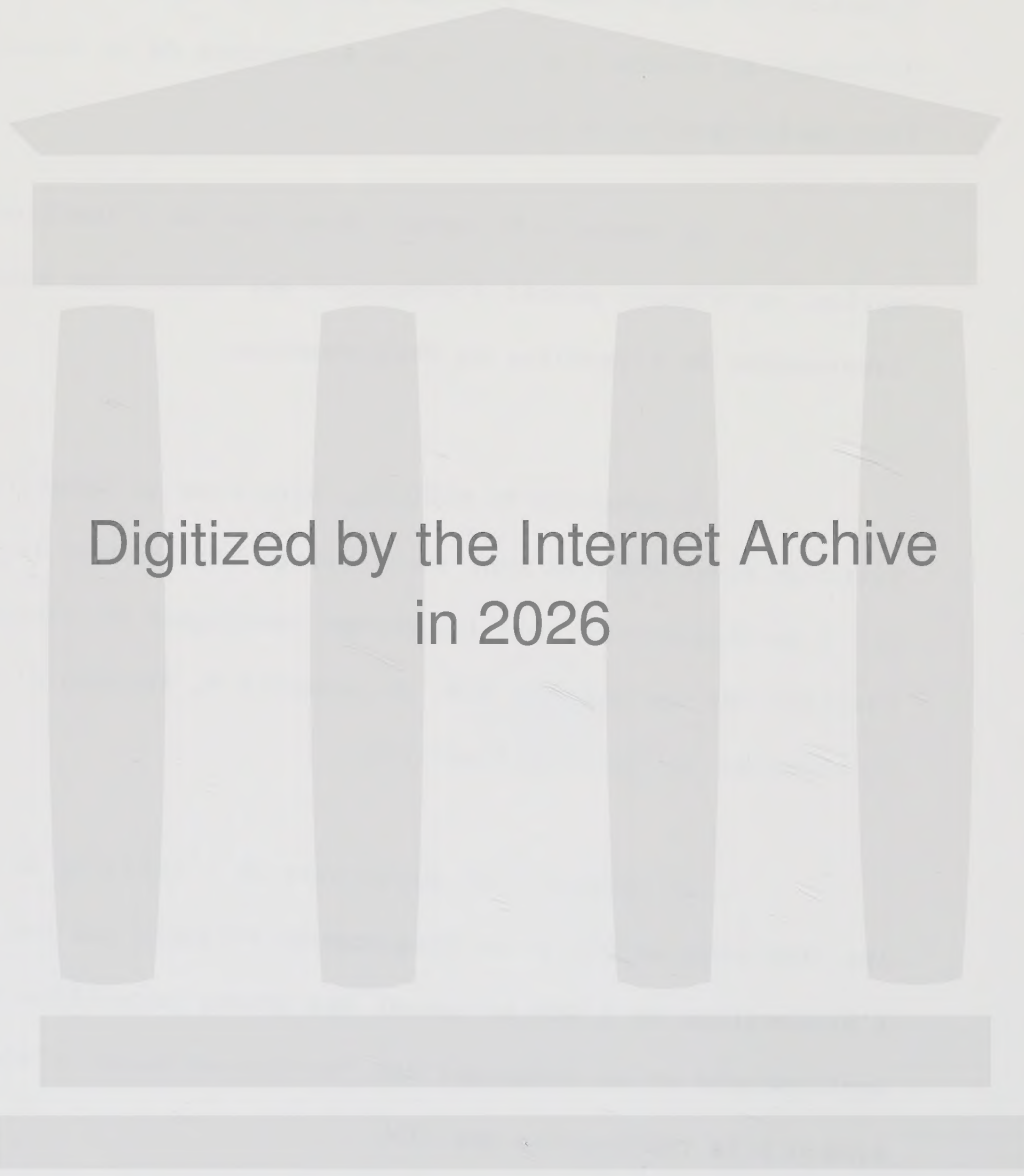
Je remercie M. SIESTRUNCK, Professeur à l'Université de PARIS VI d'avoir bien voulu accepter la présidence du Jury, ainsi que M. VIGNES, Professeur à l'Institut de Programmation, M. COURTILLOT, Directeur du Département Informatique de la Faculté d'Orsay et M. LOMBARD Directeur du Centre d'Etudes et de Recherches de la Machine Outil pour leur participation au Jury.

Je remercie M. ARSAC, Directeur de l'Institut de Programmation, de m'avoir permis d'effectuer mes recherches dans le cadre du laboratoire de l'Institut de Programmation.

Je remercie M. SUCHARD, Directeur du Laboratoire de l'Institut de Programmation pour avoir dirigé mes travaux de recherche et mis à ma disposition tous les moyens techniques du laboratoire pour réaliser une application ULM. Je remercie M. SUCHARD d'avoir bien voulu accepter la tâche de Rapporteur.

Je remercie les Assistants de l'Institut de Programmation, les étudiants en 'DEA' et en 'Programmeur d'Etudes' qui ont contribué à l'élaboration de l'ULM en menant des études parallèles sur diverses applications et en élaborant des "outils software" d'aide à la conception et à la réalisation des ULM.

Je remercie le personnel du laboratoire qui m'a aidé au cours de mes recherches pratiques.



Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41543732_0027

N° d'enregistrement
au C.N.R.S.

THESE DE DOCTEUR-INGENIEUR :
Mathématiques

présentée
A L'UNIVERSITE PARIS VI

par
M. BOSSUET Gérard

pour obtenir
1e GRADE DE DOCTEUR INGENIEUR

SUJET de la THESE

ETUDE ET REALISATION D'OPERATEURS PERMETTANT PAR LEUR
ASSEMBLAGE LA CONSTRUCTION DE STRUCTURES MICROPROGRAMMEES GE-
NERALES OU SPECIALISEES. ETUDE PRATIQUE.

Soutenue le :

JUIN 1971

devant la Commission d'Examen

MM. R. SIESTRUNCK

J. VIGNES

J. LOMBARD

J. SUCHARD

M. COURTILLOT

Président

Examineurs

N° d'enregistrement
au C.N.R.S.

THESE DE DOCTEUR-INGENIEUR :
Mathématiques

présentée
A L'UNIVERSITE PARIS VI

par
M. BOSSUET Gérard

pour obtenir
1e GRADE DE DOCTEUR INGENIEUR

=====

SUJET de 1a THESE

ETUDE ET REALISATION D'OPERATEURS PERMETTANT PAR LEUR
ASSEMBLAGE LA CONSTRUCTION DE STRUCTURES MICROPROGRAMMEES GE-
NERALES OU SPECIALISEES. ETUDE PRATIQUE.

Soutenue le :

JUIN 1971

devant la Commission d'Examen

MM. R. SIESTRUNCK

J. VIGNES

J. LOMBARD

J. SUCHARD

M. COURTILLOT

Président

Examineurs

Je remercie M. SIESTRUNCK, Professeur à l'Université de PARIS VI d'avoir bien voulu accepter la présidence du Jury, ainsi que M. VIGNES, Professeur à l'Institut de Programmation, M. COURTILLOT, Directeur du Département Informatique de la Faculté d'Orsay et M. LOMBARD, Directeur du Centre d'Etudes et de Recherches de la Machine Outil pour leur participation au Jury.

Je remercie M. ARSAC, Directeur de l'Institut de Programmation, de m'avoir permis d'effectuer mes recherches dans le cadre du laboratoire de l'Institut de Programmation.

Je remercie M. SUCHARD, Directeur du Laboratoire de l'Institut de Programmation pour avoir dirigé mes travaux de recherche et mis à ma disposition tous les moyens techniques du laboratoire pour réaliser une application ULM. Je remercie M. SUCHARD d'avoir bien voulu accepter la tâche de Rapporteur.

Je remercie les Assistants de l'Institut de Programmation, les étudiants en 'DEA' et en 'Programmeur d'Etudes' qui ont contribué à l'élaboration de l'ULM en menant des études parallèles sur diverses applications et en élaborant des "outils software" d'aide à la conception et à la réalisation des ULM.

Je remercie le personnel du laboratoire qui m'a aidé au cours de mes recherches pratiques.

O-O

Le présent document appartient à une série de publications faisant la synthèse des travaux effectués sur les Unités Modulaires Microprogrammées, au Laboratoire de l'Institut de Programmation, en 1969, 1970, 1971, sous la direction de M. SUCHARD.

O-O

- S O M M A I R E -

A. <u>INTRODUCTION.</u>	<u>PAGE 1</u>
B. <u>DEFINITION ET STRUCTURE DES UNITES LOGIQUES MODULAIRES MICROPROGRAMMEES.</u>	<u>PAGE 4</u>
B1. Définition.	page 4
B2. Définition Hardware d'une ULM.	page 6
B3. Structure de base	page 6
C. <u>MICROPROGRAMMATION</u>	<u>PAGE 12</u>
C1. Définition d'une microinstruction.	page 12
C2. Structure du microprogramme.	page 14
C3. Ecriture symbolique d'un microprogramme	page 20
C4. Choix d'une matrice de microprogrammation	page 27

D. <u>ETUDE ET CHOIX DES COMPOSANTS</u>	<u>PAGE 39</u>
D1. Choix d'un type de composants.	page 39
D2. Choix d'une famille de circuits intégrés.	page 41
D3. Conclusion.	page 55
E. <u>REALISATION DES OPERATEURS ET ENSEMBLES ULM</u>	<u>PAGE 57</u>
E1. Catalogue des circuits intégrés.	page 58
E2. Etude hardware des différents Opérateurs et Ensembles.	page 60
E3. Etude des signaux de commande - Diagramme des temps.	page 96
E4. Réalisation d'une ULM.	page 105
E5. Conclusion.	page 110
F. <u>ECRITURE D'UN SIMULATEUR ULM</u>	<u>PAGE 111</u>
F1. Choix d'un langage de programmation.	page 111
F2. Organisation du simulateur.	page 112
F3. Utilisation du simulateur de microprogramme ULM.	page 112

G. REALISATION D'UNE STRUCTURE ULM

PAGE 116

G1. Etude du système.	page 116
G2. Ecriture du microprogramme.	page 117
G3. Etude du microprogramme.	page 119
G4. Etude de la matrice de microprogrammation.	page 121
G5. Réalisation de l'ULM.	page 122
G6. Conclusion.	page 131

H. ULM APPLICATION : GENERATEUR DE VECTEURS

PAGE 133

H1. Définition.	page 133
H2. Algorithme de tracé de vecteur en relatif incrémental.	page 134
H3. Définition du périphérique commandé.	page 141
H4. Définition de la commande de l'ULM.	page 142
H5. Définition des éléments d'information.	page 143
H6. Mécanisme des échanges.	page 145

H7. Ecriture du microprogramme théorique.	page 148
H8. Choix des composants de l'ULM.	page 167
H9. Etude du microprogramme.	page 176
H10. Microprogrammation réelle.	page 182
H11. Comparaison des deux solutions.	page 202
H12. Implantation de l'ULM.	page 203

-O-O-O-O-

A. INTRODUCTION

L'étude des petits systèmes a montré la nécessité de réaliser des unités bon marché présentant à la fois la souplesse de programmation des petits calculateurs et la spécialisation des contrôleurs cablés.

L'étude des ULM a été orientée vers la définition, d'une part d'une structure de base dans laquelle viendront s'insérer un ensemble d'éléments standards, d'autre part d'une microprogrammation souple adaptée à la gestion d'une telle structure.

Notre but est d'effectuer une étude Hardware de la structure ULM et de ses différents éléments, en essayant de développer plus particulièrement les points suivants :

- Etude et réalisation d'une matrice de microprogrammation peu coûteuse, adaptée aux structures modulaires.
- Choix d'une famille de composants pour réaliser une structure ULM.
- Réalisation d'Opérateurs et d'Ensembles.
- Etude d'une application destinée à montrer les différentes étapes de l'étude et de la réalisation d'une ULM particulière.

L'étude de l'application aura pour conséquences la réalisation d'une unité spécialisée : un générateur de vecteurs pour table traçante, à déplacement relatif incrémental, commandée à distance par ligne télégraphique.

O-O-O-O-O-O-O

Signification des lettres , chiffres et abréviations utilisés

1	Sous ensemble 1 , travaillant au niveau du bit
4	Sous ensemble 4 , travaillant sur des informations de 4 bits
A	Accumulateur - Boite à opérations
B	Bascule réelle - 1 bit
C	Signal de commande élaboré par la matrice de microprogrammation
D	Débordement - Donnée
E	Entrée par rapport à l'élément considéré
G	Bascule de gadget
H	Horloge - commande séquentielle
J	Bascule de saut
K	Broche de connecteur - Interface entre deux opérateurs
L	Ensemble de 4 bits appartenant à un ensemble de 4 n bits accessibles 4 par 4 .
M	Registre d'adressage de la Matrice de microprogrammation
MM	Matrice de microprogrammation
N	Signal de validation conditionnelle d'un opérateur
P	Périphérique
R	Registre réel de 4 bits
S	Sortie par rapport à l'élément considéré
V	Bascule appartenant à un ensemble de n bascules , accessibles 1 à 1 .
X	Adressage d'un opérateur à structure matricielle
Y	Adressage " " " " "

B. DEFINITION ET STRUCTURE DES UNITES LOGIQUES MODULAIRES
MICROPROGRAMMEES.

B1. Définition

Une ULM est une unité destinée à gérer les échanges entre une unité centrale capable d'effectuer des travaux complexes et un ensemble de périphériques.

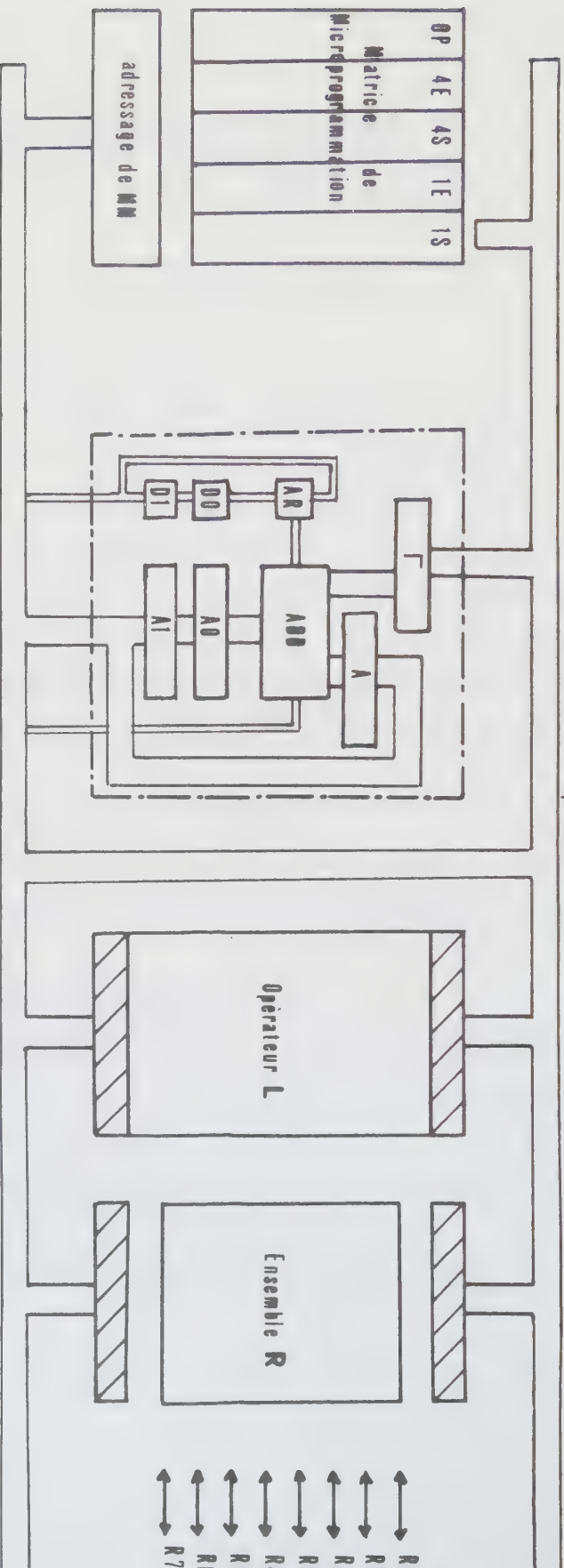
Le cycle machine d'une ULM dépend de la matrice de microprogrammation. Sa valeur minimale est de l'ordre de 200ns, temps court par rapport aux cycles des Unités auxquelles l'ULM est connectée. Donc la longueur des éléments d'information traités dans un cycle machine, est indépendante de la longueur des éléments d'information à traiter pour les échanges.

Cependant, la longueur du microprogramme est proportionnelle au rapport de la longueur des éléments d'information à traiter, à la longueur des éléments d'information traités pendant un cycle machine.

Pour une application courante, une structure 4 bits est bien adaptée. Cependant, ce choix n'exclut pas, pour des applications très particulières, une structure P bits, $P \neq 4$.

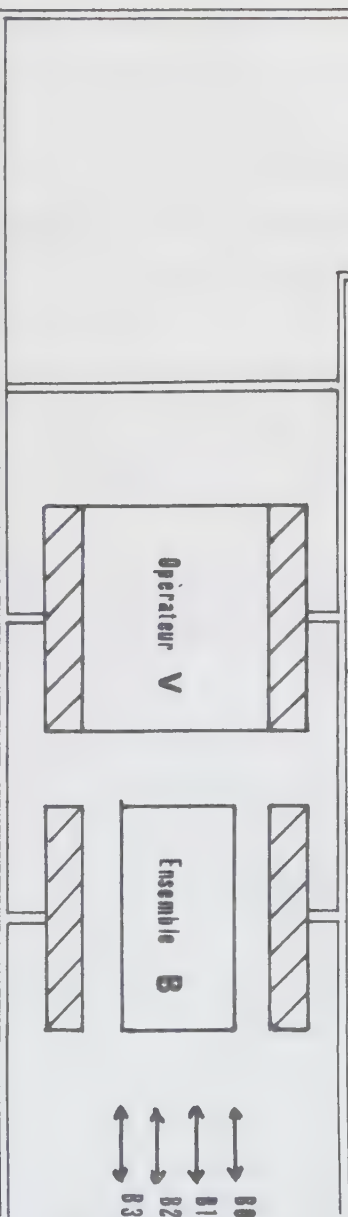
Une ULM est une machine virtuelle 4N bits, conçue à partir d'une machine réelle 4 bits microprogrammée.

BUS entrée Boite à Opérations - 4E



BUS sortie Boite à Opérations - 4S

BUS Report - 1E



BUS Débordement - 1S

B2. Définition Hardware d'une ULM

Chaque élément de la structure de base est un ensemble de circuits logiques ayant une fonction bien déterminée. Il existe deux types d'éléments dont la seule différence se situe au niveau Hardware :

- un Opérateur possède un nombre fixe de circuits. C'est un élément standard. (Boîte à Opérations)
 - un Ensemble possède un nombre variable de circuits, en fonction du problème à résoudre. (Ex : ensemble de registres réels de dialogue avec l'extérieur).
- Un élément d'un Ensemble peut être un Opérateur.

Une ULM est un assemblage d'Opérateurs et d'Ensembles défini en fonction d'un problème donné et géré par microprogramme.

B3. Structure de base

L'unité s'organise autour de l'Opérateur Traitement.

La Boîte à Opérations est un accumulateur construit à partir d'un additionneur parallèle 4 bits.

<u>Entrée</u>	Opérande 1	Accumulateur
	Opérande 2	appartenant à la structure
	Report Initial	
<u>Sortie</u>	Accumulateur	
	Débordement	

La structure de base comportera donc un ensemble de quatre bus :

<u>Bus hexadécimal</u>	- Entrée Boîte à Opérations , Opérande 2
<u>Bus hexadécimal</u>	- Sortie Boîte à Opérations , Accumulateur
<u>Bus binaire</u>	- Entrée Boîte à Opérations , Report initial
<u>Bus binaire</u>	- Sortie Boîte à Opérations , Débordement.

Ces quatre bus définissent deux demi-structures , hexadécimale et binaire symétriques, à l'intérieur desquelles les Ensembles et les Opérateurs auront des rôles analogues.

Pour assurer les échanges avec le monde extérieur et le traitement des données, la structure de base comporte 4 types d'Opérateurs et d'Ensembles:

- Stockage : Registres et Bascules..
- Traitement : Boîte à Opérations.
- Echanges : Bus.
- Commande : Matrice de Microprogrammation.

1. Stockage :

Entre les deux bus de chaque demi-structure, il faut assurer la fonction stockage pour :

- Dialoguer avec le monde extérieur.
- Exécuter le traitement.

La fonction Dialogue dépend essentiellement du monde extérieur. Elle sera donc assurée par des Ensembles composés d'éléments réels, accessibles en parallèle pour le monde extérieur et un à un pour la Boîte à Opérations.

- Demi-structure hexadécimale : Ensemble R de registres réels de 4 bits.
- Demi-structure binaire : Ensemble B de bascules de dialogue.

La fonction Traitement est interne à l'ULM. Elle sera donc assurée par des Opérateurs composés d'éléments accessibles un par un pour la Boîte à Opérations. Nous avons choisi des Opérateurs contenant 16 éléments :

- nombre hexadécimal permettant d'utiliser d'une manière simple l'adressage d'un élément quelconque de la structure par un autre élément de la structure.
- raisonnable pour la plupart des applications.
- facilement réalisable au point de vue hardware.

- Demi-structure hexadécimale : Opérateur L de 16 registres "locaux".
- Demi-structure binaire : Opérateur V de 16 bascules "virtuelles".

La structure étant modulaire, il est nécessaire de pouvoir inhiber chaque opérateur.

Remarque :

La structure n'a pas de mémoire. Cette dernière doit être considérée comme un périphérique.

2. Traitement

La Boîte à Opérations est un additionneur parallèle, donc le premier opérande est soit :

- Le contenu de l'accumulateur.
- zéro.

Le second opérande est soit :

- Le contenu d'un registre R ou L.
- Le contenu de l'accumulateur.
- Une constante hexadécimale K.
- Le contenu concaténé d'un ensemble de 4 bascules B.
- zéro.

Le second opérande peut être successivement :

- masqué : par une constante K.
par le contenu d'un registre R ou L.
à la fois par les deux.

Le masque réalise la fonction "ET" en logique positive.

- complémenté.

Un adressage de tous les registres permet de choisir comme second opérande n'importe quel registre de la structure.

L'accumulateur AO est un registre transparent et doit être doublé par un registre A1 pour servir de premier opérande. A chaque cyle la recopie de l'accumulateur dans sa doublure permet de faire une opération supplémentaire à la demande de l'utilisateur : par exemple un décalage à droite, une image miroir, une complémentation....

Les opérations addition et complémentation peuvent être rendues conditionnelles par l'intermédiaire de deux bascules dont les états indiqueront les opérations à effectuer. Le résultat de l'addition est un nombre hexadécimal et un bit de débordement, l'entrée Report Initial de l'additionneur permettant à l'unité de travailler sur format variable.

3. Echange

Un bus se décompose en un ensemble de barrières appartenant aux opérateurs et mis en parallèle par câblage.

Chaque barrière possède une commande de validation.

4. Commande

La fonction commande est assurée par la matrice de microprogrammation. Celle-ci élabore les commandes élémentaires réelles. Le mécanisme d'adressage de la matrice de microprogrammation et l'exploitation des commandes se rapportent à la microprogrammation.

0-0-0-0-0-0-0-0-0-0

C. MICROPROGRAMMATION

C1. Définition d'une microinstruction

Une microinstruction est un ensemble de commandes élémentaires indépendantes, exécutables pendant un cycle machine.

C1a) Définition des commandes élémentaires :

Les commandes élémentaires sont élaborées par la matrice de microprogrammation. Leur élaboration est soumise à un certain nombre d'hypothèses, que nous allons rappeler.

Hypothèses :

- Les commandes élémentaires :

1. sont élaborées jusqu'à la fin du cycle machine.
2. gardent la même signification pendant tout le cycle.
3. sont indépendantes de toute opération effectuées dans la même microinstruction.
4. sont directes.
5. sont spécifiques d'une seule opération.
6. sont exécutables simultanément.

- Toute commande de transfert ou de traitement est relative à la Boîte à Opérations.

C1b) Définition des champs :

Le microprogramme gère la Boîte à Opérations, les échanges avec l'extérieur, et commande les bus de la structure.

Une microinstruction sera donc divisée en 5 champs :

- 1 champ opération OP, élaborant les commandes de gestion de la Boîte à Opérations, du microprogramme (sauts) et les ordres d'entrée-sortie.
- 4 champs de commande des 4 bus de la structure : 4E, 4S, 1E, 1S :

E : entrée par rapport à la boîte à opérations

S : sortie par rapport à la boîte à opérations

4 : demi-structure hexadécimale

1 : demi-structure binaire

Hypothèse : Chaque champ n'adresse qu'un seul opérande.

1. Entrée dans la BØ : CHAMP 4E

Selectionne le registre dont le contenu servira de second opérande à l'additionneur.

2. Sortie de la BØ : CHAMP 4S

Selectionne le registre dans lequel sera rangé l'accumulateur contenant le résultat de la microinstruction précédente.

3. Rangement du débordement : CHAMP 1S

Sélectionne une bascule B ou V dans laquelle sera rangée la valeur du débordement de l'accumulateur.

Si le champ opération contient la commande "constante", la valeur de ce dernier est prise dans ce champ

4. Test des bascules : CHAMP 1E

Ce champ a deux rôles :

- ajouter à l'accumulateur la valeur du contenu d'une bascule B ou V.
- tester B ou V, afin de valider ou non la microinstruction en cours.

C2. Structure du microprogramme

C2a) Structure d'une microinstruction :

Pendant un cycle machine, les opérations à effectuer sont :

1. Commande des bus hexadécimal et binaire d'entrée. dans la boîte à opérations.
2. Commande de la boîte à opérations, commande de l'accumulateur pour mémoriser le résultat du traitement.
3. Recopie de AO dans A1.
4. Commande des bus hexadécimal et binaire de sortie de la boîte à opérations.
5. Commande d'opérations d'échange avec l'extérieur.
6. adressage de la microinstruction suivante.

Le doublage de A0 permet d'effectuer simultanément les opérations 1 et 4 d'une part, 3 et 6 d'autre part, si le registre dont le contenu est utilisé comme deuxième opérande et le registre dans lequel est rangé l'accumulateur sont différents.

L'opération 5 est indépendante des autres.

Afin de diminuer le cycle de la machine, nous décomposerons donc une microinstruction en deux temps élémentaires :

- temps 1 : Opérations 1, 2, 4, 5.
- temps 2 : Opérations 3, 6.

Ce découpage entraîne pour la structure du microprogramme deux contraintes :

- Un registre utilisé comme opérande 2 de la boîte à opérations ne peut servir pour stocker A1 dans la même microinstruction.
- Le résultat stocké est celui de la microinstruction précédente.

C2b) Exécution d'une microinstruction

Chaque microinstruction peut être rendue conditionnelle par lecture d'une bascule (B ou V). Le test de cette bascule sera donc inclus dans le champ 1E qui gère les éléments binaires en lecture.

Le microprogramme peut élaborer n'importe quelle constante hexadécimale. La constante est un opérande. Elle doit donc être générée à partir d'un champ de commande de bus.

l'ar hypothèse :

- Tout opérande en entrée de la boîte à opérations doit pouvoir être masqué par une constante.
- Tout résultat de microinstruction doit pouvoir être rangé sans condition.
- Toute microinstruction doit pouvoir être rendue conditionnelle.

Le rangement du débordement dans une bascule est une opération moins utilisée que les précédentes et peut se faire avec retard si le contenu de D1 n'est pas détruit systématiquement à chaque microinstruction.

Le chargement d'une constante dans AO ou le masque d'un opérande en l'absence de commande A, opérations les plus fréquentes, donnent un débordement nul.

Donc, la constante sera élaborée par le champ 1S, si le contenu de la bascule D1 n'est pas détruit lorsque le champ OP contient la commande K. Il n'est pas nécessaire de créer un champ spécial.

Si un opérande est utilisé en champ 4E simultanément, nous aurons masque de l'opérande par la constante avant entrée dans la boîte à opérations.

La structure de saut définie est suffisamment souple pour simplifier le Hardware ULM , en rendant chaque registre adressage indépendant. La matrice de microprogrammation sera donc ainsi divisée en blocs de 16 pages de 16 lignes.

Chaque registre MO, M1, M2 est en relation avec le bus de sortie de la boîte à opérations.

MO, M1, M2 sont des registres transparents. Il est donc nécessaire de les doubler pour adresser une microinstruction et préparer l'adresse de la suivante.

A chaque microinstruction MO est recopié dans MMO :
il contient soit $(MMO) + 1$
soit $(A1)$

A chaque microinstruction dans laquelle figure la commande "saut" en champ OP, M1 et M2 sont recopiés dans MM1, MM2 après exécution de la microinstruction.

L'adresse de la microinstruction suivante est soit :

$(MM2)/(MM1)/(MMO+1)$: dans le cas courant.
 $(MM2)/(MM1)/(MO)$: si une valeur a été introduite par la microinstruction dans MO.
 $(M2)/(M1)/(MMO+1)$: en cas de saut sans chargement de MO.
 $(M2)/(M1)/(MO)$: en cas de saut avec chargement de MO.

Tout élément de la structure est accessible par adressage à partir d'éléments réels de la structure. Le microprogramme, bien que câblé une fois pour toutes, pourra donc effectuer des opérations différentes pour une même microinstruction.

Dans le champ OP, les opérations d'addition et de complémententation peuvent être rendues conditionnelles, par test de deux bascules réelles.

Une microinstruction peut donc être :

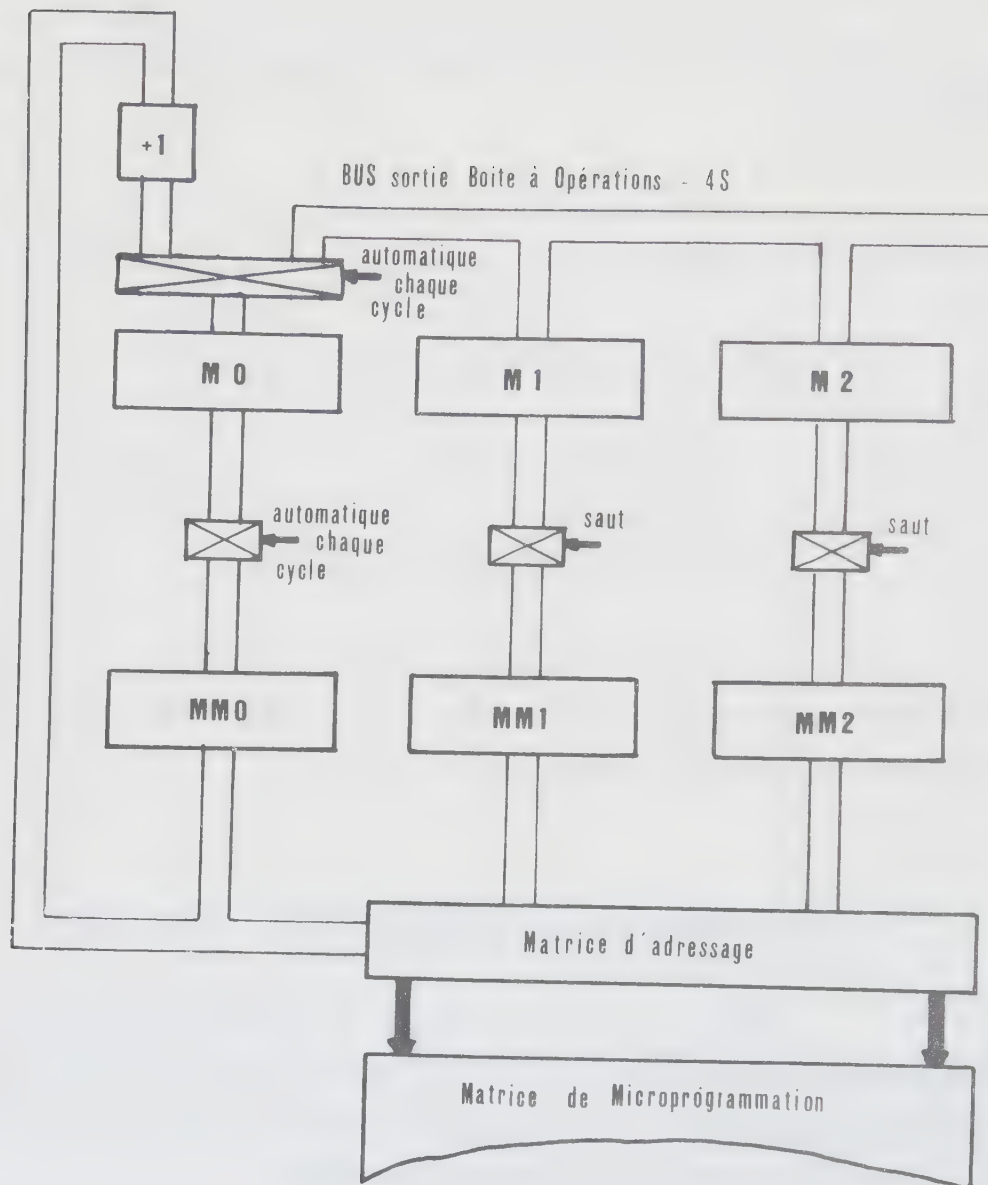
conditionnelle en champ OP ou contenir une commande d'adressage, et ne pas être totalement définie avant exécution, en champ 4E, 4S, 1E, 1S.

L'écriture d'un microprogramme ULM présente une très grande souplesse. L'exécution se fait par adressage successif des différentes microinstructions et exécution éventuelle des commandes.

C2c) Enchaînement des microinstructions : (Fig. C1)

Exécuter un microprogramme, c'est d'abord élaborer les commandes relatives à une microinstruction, et ensuite déterminer quelle sera la microinstruction suivante. Du choix du mode d'adressage de la microinstruction dépend la souplesse d'utilisation du microprogramme.

Pour adresser la matrice de microprogrammation, nous choisirons 3 registres réels de la structure hexadécimale, M0, M1, M2. Une matrice contiendra au maximum 4096 microinstructions.



Adressage de la matrice de Microprogrammation

C3. Ecriture symbolique d'un microprogramme

C3a) Adresse de la microinstruction

se compose de trois chiffres hexadécimaux :

numéro de ligne M0

numéro de page M1

numéro de bloc M2

C3b) Notations

h : chiffre hexadécimal 0 à F

hh : nombre hexadécimal 00 à FF

n : chiffre décimal 0 à 2

. : rien

k : chiffre hexadécimal contenu dans un registre réel
de la machine, utilisé pour l'adressage.

C3c) Microinstruction

Une microinstruction définit les opérations élémentaires
exécutables simultanément dans la machine. Les notations utilisées pour définir
les commandes correspondantes aux 5 champs sont les suivantes :

C3c1) Champ opération OP : ensemble de 2 à 8 lettres.

1ère lettre : définit le second opérande.

K : constante définie dans le champ 1S.

non K : opérande défini dans le champ 4E.

2ème, 3ème, 4ème et 5ème lettres : définissent les commandes de la boîte à opérations.

M : masque "ET" entre A1 et le deuxième opérande.

. : l'opérande est pris directement en vraie valeur.

C : l'opérande est complémenté.

D : l'opérande est complémenté ou non suivant le contenu d'une bascule réelle.

A : addition binaire de l'opérande et du contenu de l'accumulateur avec report initial éventuel, suivant le champ 1E.

B : addition ou non suivant le contenu d'une bascule réelle.

. : l'entrée dans l'additionneur de l'accumulateur est inhibé, donc addition de 0, avec report éventuel, suivant 1E.

G(h) : indication de commande d'une partie supplémentaire de la boîte à opérations dont la fonction est définie par l'utilisateur.

Il est possible de définir 16 fonctions. Si la 4ème lettre n'est pas G, la partie supplémentaire est non commandée ou inexistante.

6ème lettre : S : saut à la microinstruction dont l'adresse est contenue dans les registres M, après exécution de la microinstruction contenant S.

7ème lettre : P(h) : définit les opérations d'échanges avec l'extérieur, échange avec un périphérique en mode parallèle, par l'intermédiaire des registres R.

8ème lettre : H(h) : définit une opération hardware cablée pour un utilisateur.

Remarques

- cas de la constante K :

Lorsqu'un opérande est défini dans le champ 4E et une constante dans le champ 1S, on aura une opération de masque de l'opérande par la constante avant passage dans le complémenteur.

L'opération ainsi effectuée est un "ET" .

	f	v
f	f	f
v	f	v

K : constante.(4E)

- masque :

Fonction "ET" entre le 1er et le 2ème opérande

M : (A1).((4E))

- association de deux commandes :

KM : (A1).((4E)).Cte

Opération "ET" entre le premier opérande, l'opérande défini dans le champ 4E et la constante définie dans le champ 1S.

Les commandes K et M agissent sur le débordement de l'additionneur.

Le débordement sera positionné en fonction de l'opération "OU" entre celui de la microinstruction en cours et celui de la précédente.

	f	v
f	f	v
v	v	v

C3c2) Champ 4E :

Le second opérande de la boîte à opérations est :

. : non défini et pris égal à F

AE : contenu de l'accumulateur A1

VE(h) : contenu de 4 bascules consécutives

BE(h) : contenu de 4 bascules concaténées

L(hh) : contenu du registre L(hh)

R(hh) : contenu du registre R(hh)

IR(h) : contenu du registre R(hk)

IL(h) : contenu du registre L(hk)

C3c3) Champ 4S :

Le contenu de A1 est rangé dans :

. : aucun registre

R(hh) : le registre R défini

L(hh) : le registre L défini

M(n) : le registre M défini

VS(h) : 4 bascules V de l'opérateur V

BS(h) : 4 bascules B de l'ensemble B
IR(h) : le registre R(hk)
IL(h) : le registre L(hk)

C3c4) Champ 1E

. : rien
RU : report initial de l'additionneur égal à 1
RD : report direct du débordement généré par la
microinstruction précédente
RV(hh) : le contenu de la bascule V(hh) ou B(hh) sert
RB(hh) de report initial à l'additionneur
IRV(h) : Le contenu de la bascule V(hk) ou R(hk) sert
IRB(h) de report initial à l'additionneur
TV(hh) : test du contenu de la bascule V(hh) ou B(hh)
TB(hh) pour valider la microinstruction
1 : validation
0 : suite en séquence
TD : test sur le contenu du débordement pour vali-
der la microinstruction le contenant
ITV(h) : test du contenu de la bascule V(hk) ou B(hk)
ITB(h)

C3c5) Champ 1S

. : rien
B(hh) : rangement du débordement généré par la micro-
V(hh) instruction précédente dans la bascule V ou B
adressée.

U L M

DE	IN	NO	OP	4E	4S	7E	1S	CONCERNÉES
		0						
		1						
		2						
		3						
		4						
		5						
		6						
		7						
		8						
		9						
		A						
		B						
		C						
		D						
		E						
		F						

$\left. \begin{array}{l} IB(h) \\ IV(h) \end{array} \right\} : \text{Rangement dans la bascule } B(hk) \text{ ou } V(hk)$

0, 1, 2, ...9, A, ...F, constante hexadécimale

R E C A P I T U L A T I O N

M2	M1	M0	OP	AE	IR	IR	IS
			. ou A ou B	L(hh)	L(hh)	RV(hh)	V(hh)
			. ou C ou D	R(hh)	R(hh)	RB(hh)	R(hh)
			. ou K	AE	P(n)	RU	TV(h)
			. ou M	VE(h)	VS(h)		
			. ou S	BE(h)	BS(h)	RD	TB(h)
			. ou G	IR(h)	IR(h)	TV(hh)	h
			. ou P(h)	IL(h)	II(h)	TB(hh)	.
			. ou H(h)	.	.	TD	
						ITV(h)	
						ITB(h)	
						IRV(h)	
						IRB(h)	

C3d. Format des feuilles de microprogrammation

Pour écrire un microprogramme, le format choisi est celui de la figure C2:

C4. Choix d'une matrice de microprogrammation

C4a) Définition :

Une matrice de microprogrammation est une grille pour laquelle a été défini un moyen de couplage entre chaque ligne et chaque colonne.

C4b) Critères de choix :

En fonction d'un problème donné, il n'existe pas une ULM, mais un ensemble d'ULM construites en petites séries. La matrice de microprogrammation d'une structure ULM devra donc présenter les avantages suivants :

- Etre construite à partir d'un élément de couplage permettant une construction standard de la matrice.
- Etre bon marché.
- Etre rapide pour certaines applications.
- Etre fiable, facilement réparable et éventuellement modifiable.

Ces critères éliminent toute solution du type matrice intégrée, où les éléments de couplage sont réalisés à la demande, lors du dépôt de la dernière couche de métallisation du circuit intégré.

L'encombrement de la matrice de microprogrammation n'étant pas un critère fondamental, il est préférable de choisir un montage à éléments de couplage distincts et interchangeables.

C4c) Choix d'un élément de couplage

Il existe actuellement un grand nombre de techniques pour construire une matrice de microprogrammation.

Hypothèse

Afin de simplifier au maximum le hardware, les sorties de la matrice de microprogrammation devront être permanentes pendant un cycle.

Cette hypothèse élimine tous les systèmes de couplage temporaires tels que, capacités , selfs, tores, nécessitant une mémorisation à la sortie.

Les seuls composants de couplage qui répondent actuellement aux critères posés sont la résistance et la diode.

Le choix entre l'un des deux éléments de couplage résultera d'une comparaison entre les avantages et les inconvénients d'une matrice à résistances et d'une matrice à diodes.

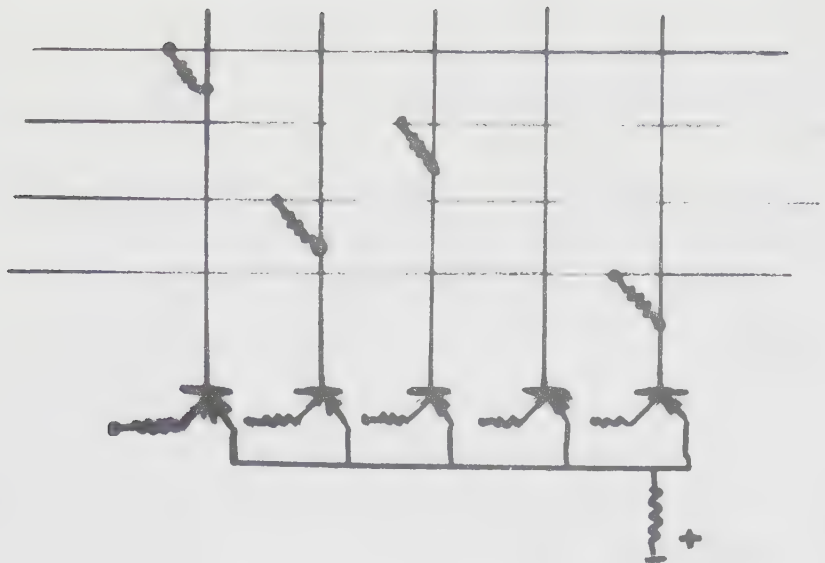


Fig C3. Matrice à résistances

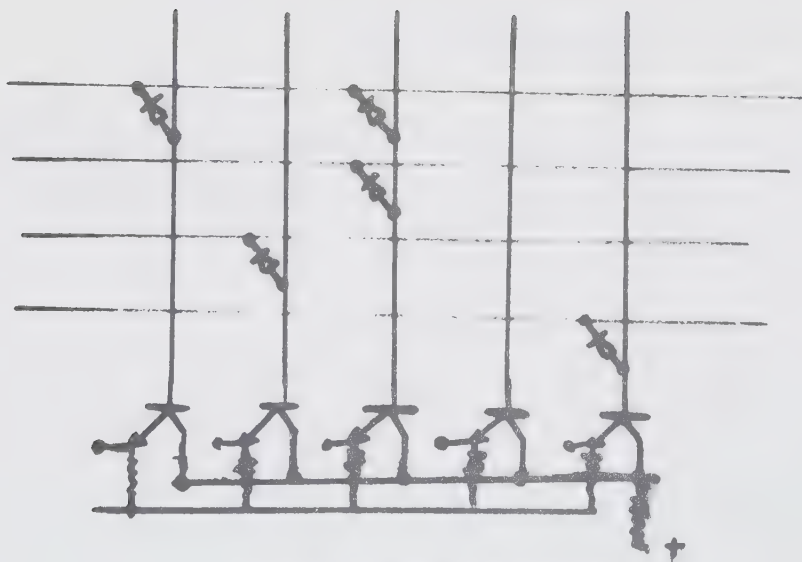


Fig C4. Matrice à diodes

Avantages et inconvénients d'une matrice à résistances (Fig. 03)

Avantages :

- le temps de réponse de la matrice ne dépend que des transistors d'entrée et de sortie. Il peut être très court (20ns)
- la modification du microprogramme est aisée
- l'alimentation de la matrice est constante
- la résistance est un composant bon marché. Cependant cet avantage tend à disparaître et même à devenir un inconvénient par rapport à l'évolution des prix des diodes de commutation. Le prix d'une résistance n'a pratiquement pas varié, alors que les diodes sont maintenant beaucoup moins chères que les résistances.

Inconvénients :

- les microinstructions ne sont pas indépendantes les **unes des autres**. La variation de tension d'une colonne adressée dépend du nombre de microinstructions adressant la même colonne.
- La variation de niveau d'une colonne est inversement proportionnelle au nombre de microinstructions couplées à cette colonne. Le nombre de microinstructions est donc limité supérieurement.
- La variation d'impédance d'une colonne est inversement proportionnelle au nombre de microinstructions couplées à cette colonne. Pour le transistor de sortie, cette variation a une borne supérieure. Le nombre de microinstructions est donc limité inférieurement.

D'après les essais que nous avons effectués, les bornes inférieure et supérieure sont très proches (7 - 8 dans un cas, 12-16 dans un autre). Ceci pose des problèmes d'équilibrage, quelquefois impossibles à résoudre dans le cas d'une matrice à champs linéaires.

Une structure par demi-champs, à organisation matricielle, diminue beaucoup les problèmes d'équilibrage, mais en contre-partie nécessite deux fois plus d'éléments de couplage.

- La matrice de microprogrammation doit être une matrice non binaire, à champs. (Montage différentiel des transistors de sortie)
- Chaque microinstruction doit adresser une colonne dans chaque champ. Le nombre de transistors de sortie ne dépend donc pas des états du microprogramme, mais du nombre de microinstructions.
- Le nombre d'éléments de couplage est constant :

M microinstructions	}	nombre de résistances : $M \times P$
P champs		

Avantages et inconvénients d'une matrice à diodes (Fig. C4)

Avantages :

- Chaque microinstruction est indépendante des autres. Une colonne de la matrice peut donc être couplée à un nombre quelconque de microinstructions.
- La tension d'une colonne ne prend que 2 valeurs :
$$\begin{cases} V_{cc} & \text{si elle n'est pas adressée} \\ 0 & \text{si elle est adressée.} \end{cases}$$

Les transistors de sortie peuvent donc être indépendant et travailler sur des niveaux équivalents à ceux utilisés pour les circuits intégrés de la structure.
- Le nombre de diodes par microinstruction n'est pas fixe. Nous aurons la possibilité d'obtenir des sorties linéaires ou binaires.
- Le nombre d'éléments de couplage et d'éléments de sortie dépend de la nature du microprogramme et non du nombre de microinstructions.

Inconvénients :

- Les diodes, en inverse, ont une impédance très grande et la matrice est lente.
- L'alimentation en courant de la matrice est variable.
Elle dépend du nombre de diodes couplées par micro-instruction. Nous verrons, dans le paragraphe suivant, qu'une matrice binaire peut demander une alimentation variable très importante.

Pour une structure modulaire, les diodes présentent l'avantage de ne pas nécessiter un équilibrage de la matrice, mais sont lentes et nécessitent une alimentation variable dépendant, en binaire, de l'ULM. L'étude du paragraphe suivant permettra un choix entre les deux types de composants.

C4d) Choix d'une structure de matrice de microprogrammation :

Hypothèse : Les sorties de la matrice de microprogrammation seront exploitées directement pendant le cycle où elles sont adressées.

Pour la structure de la matrice de microprogrammation, nous avons le choix entre une structure binaire et une structure non binaire à champs.

En remarquant, que la structure des opérateurs de stockage est matricielle, à adressage et commandes non binaires, nous pouvons envisager une structure à champs non binaires dans laquelle chaque champ serait partagé en deux demi-champs commandant directement les opérateurs.

Nous allons comparer les 3 types de matrice de micro-programmation pour une même application de l'ULM :

1 Boîte à Opérations	Complémenteur sans "gadget"
1 Opérateur V	
1 Opérateur L	
1 Ensemble R	8 registres réels
1 Ensemble B	4 bascules réelles

aucune opération spéciale Hardware.

1. Matrice binaire :

Champ OP	9 bits
Champ 4E	5 bits
Champ 4S	5 bits
Champ 1E	6 bits
Champ 1S	<u>5 bits</u>
Total : 30 bits	

Pour une exploitation simple, le champ OP contient autant de bits qu'il y a de commandes élémentaires. En réalité, dans un micro-programme , les commandes en champ OP sont limitées et toutes les combinaisons ne sont pas utilisées.

La sortie nécessite un décodage qui augmente le temps de retard.

Le courant à fournir est proportionnel à un nombre compris entre 0 et 30.

Le nombre de commandes élaborées, après décodage, est compris entre 0 et 14 (6+2+2+2+2).

2. Matrice non binaire - 5 champs - :

Champ OP	30	nombre raisonnable pour un micropro-
Champ 4E	16	adressage L. gramme.
	8	adressage R
	2	AE, BE
	2	IL, IR
Champs 4S	16	adressage L
	8	adressage R
	1	BS
	2	IL, IR
	3	MO, M1, M2
Champ 1E	16	adressage V report
	16	adressage V test
	4	" B report
	4	" B test
	2	RU, RD
	1	TD
Champ 1S	16	adressage V
	4	adressage B
	2	IV, IB
		la constante est générée à partir des
		16 adressages de V.

Total : 157

La sortie nécessite un recodage pour adresser les opérateurs à structure matricielle.

Le courant à fournir est proportionnel à un nombre compris entre 0 et 5.

Le nombre de commandes élaborées par la matrice est compris entre 0 et 5.

3. Matrice non binaire - 10 demi champs - structure matricielle :

Le nombre d'états dans chaque champ est le même que précédemment.

Champ OP	5 + 6 colonnes	(5x6 = 30 états)
Champ 4E	5 + 6	"
Champ 4S	5 + 6	"
Champ 1E	7 + 7	"
Champ 1S	5 + 5	"
Total	57	!

Les commandes de sortie sont directes pour les champs 4E, 4S, 1E, 1S. Un recodage est nécessaire pour le champ OP.

Le courant à fournir est proportionnel à un nombre compris entre 0 et 10.

Le nombre de commandes élaborées par la matrice est compris entre 0 et 10.

Comparaison des trois matrices :

Définitions : les coefficients définis sont des nombres entiers, les plus proches des quotients décimaux trouvés.

Coefficient de couplage CC:

rapport du nombre total de colonnes au nombre maximal de colonnes couplées par microinstruction.

Coefficient de commande Cd :

rapport du nombre de commandes nécessaire au nombre de commandes élaborées.

Coefficient d'utilisation CU :

rapport du nombre total d'états au nombre total de colonnes.

colonne	30	157	57
maximun colonnes couplées	30	5	10
commandes élaborées	14	5	10
commandes nécessaires	14	14	14
états	157	157	157
maximun éléments de couplage par page de microprogramme	480	80	160
logique de sortie	DECODAGE	COMPLEXE sauf	DIRECTE sauf
courant à fournir	29i	pour le champ CP	pour le champ CP
CC	1	31	6
Cd	1	3	1
CU	5	1	3

Les critères de choix sont :

- courant à fournir minimal, si possible constant.
- coefficient C_u d'utilisation maximal.
- coefficient C_c de couplage minimal.
- coefficient C_d de commande voisin de 1.

1. Courant :

La matrice binaire nécessite une grande quantité de courant variable sur une très grande plage. De plus, si le nombre de modules augmente, le nombre de bits de codage d'une microinstruction augmente. Donc le courant à fournir augmente.

La matrice binaire n'est pas adaptée aux structures modulaires.

2. Coefficients C_c , C_d , C_u :

La comparaison des coefficients C_c , C_d , C_u , des deux matrices non binaires montre un net avantage pour la matrice 10 champs, beaucoup moins encombrante. De plus, son coefficient de **commande** est équivalent à celui d'une matrice binaire pour un coefficient d'utilisation à peine moins élevé.

Nous choisirons une structure non binaire à demi-champs.

Remarque :

Les opérations du champ OP sont indépendantes.

L'écriture de microprogramme montre qu'un très petit nombre de combinaisons est utilisé. Il est donc très simple d'adapter la structure de ce champ à une structure linéaire. Nous économiserons à la fois des éléments logiques de sortie et un élément de couplage par microinstruction.

CONCLUSION :

- La structure de la matrice de microprogrammation sera du type non binaire :
 - linéaire en champ OP
 - matricielle en champ 4E, 4S, 1E, 1S.
- Le nombre d'éléments de couplage est limité à 9 au maximum, par microinstruction.
- L'élément de couplage choisi est la diode.

D. ETUDE ET CHOIX DES COMPOSANTS

D1. Choix d'un type de composants :

Depuis la commercialisation des circuits intégrés, il devient de moins en moins rentable de construire des ensembles logiques analogues à ceux que nous avons définis, en composants discerts - Transistors, Diodes, Résistances -

Le principal intérêt des circuits intégrés, outre une réduction de volume et un abaissement du prix de revient, est l'augmentation dans de grandes proportions de la fiabilité des systèmes.

Cet accroissement de fiabilité est du à plusieurs facteurs :

- Réduction du nombre de connections : Dans un circuit électronique, la moitié des pannes sont dues à de mauvaises connections ; soit des défauts de soudure, soit des défauts de connections entre les pastilles des semi-conducteurs et les broches de sortie. Pour un circuit intégré, les connections sont faites simultanément par évaporation d'un film métallique ; d'où élimination d'un grand nombre de causes de pannes.

- Compatibilité des composants : Pour un même circuit, tous les composants de même type sont réalisés simultanément, les caractéristiques de 2 éléments peuvent donc être très voisines et évoluer ensemble.

- Fonctionnement à faible puissance : Les éléments d'un même circuit sont très rapprochés. Les connexions courtes diminuent le problème du bruit. Les circuits intégrés peuvent donc être utilisés à un faible niveau. Ce qui entraîne une élévation de température peu importante, facilement dissipable par le boîtier.

- Réduction des dimensions : Un circuit monolithique sur plaquette de silicium à une surface inférieure à 1 cm².

Sa faible superficie le rend :

- moins sensible aux phénomènes extérieurs variables dans l'espace comme par exemple les champs magnétiques.
- moins sensible aux variations de température. Les éléments ayant les mêmes caractéristiques et étant très proches subissent les mêmes variations. Ils ne présentent donc pas de dérive de 0 pour une large gamme de température.
- plus performant au point de vue commutation, les capacités parasites étant éliminées au maximum.

Le choix d'un type de composant se ramène donc au choix d'une famille de circuits intégrés.

D2. Choix d'une famille de circuits intégrés :

Il existe actuellement sur le marché, un grand nombre de séries de circuits intégrés, définies principalement par le mode de couplage des différents étages d'un même circuit et par les éléments d'entrée du circuit.

Les séries peuvent être groupées en famille :

DCTL : Logique à transistors à couplage direct.

RTL : Logique à transistors et résistances de base.

RCTL : RTL + Capacité en parallèle avec résistances.

DTL : Logique à transistors et diodes.

TTL : Logique à transistors et transistor multiémetteur d'entrée.

D2a. Caractéristiques communes des circuits intégrés

Définitions :

Les caractéristiques des circuits intégrés dépendent, à la fois de la construction du circuit, et des éléments connectés à l'entrée et à la sortie du circuit.

Définitions :

- Tension de seuil : Tension minimum à appliquer à l'entrée d'un circuit pour le faire changer d'état. Cette tension dépend de la diode base-emetteur du transistor d'entrée, ainsi que de la configuration du circuit. Elle varie avec la température.

- Marge de bruit : Variation maximale de tension à l'entrée du circuit ne produisant aucun effet sur la logique de sortie. La marge de bruit dépend de la température. Elle dépend aussi de l'utilisation du circuit du fait des régimes transitoires générés en cours de fonctionnement.

- Vitesse de fonctionnement : Retard entre la commande d'entrée du circuit et l'obtention du niveau logique correspondant, en sortie. Ce retard est dû aux temps de commutation des transistors et plus particulièrement au temps que met le courant de base d'un transistor à charger ou décharger les éléments capacitifs intrinsèques du circuit.

- Sortance : (Fan Out), nombre de portes commandables simultanément par une même sortie.

- Température de fonctionnement : Les caractéristiques des semi-conducteurs varient avec la température, mais peuvent rester stables du point de vue logique pour une certaine gamme de température. Les circuits actuels doivent travailler :

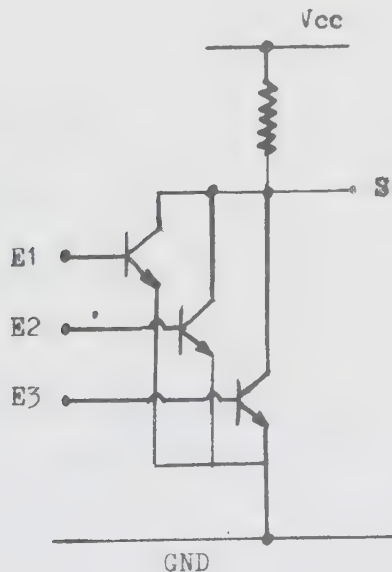
entre 0°C et 70°C	Gamme industrielle
entre 55°C et 125°C	Gamme militaire

- Dissipation : Puissance consommée par une porte ayant un facteur cyclique de 50 % (le facteur cyclique est le rapport du temps pour lequel la porte est dans l'état "0", au temps pour lequel la porte est dans l'état "1", pour une période donnée). La dissipation varie en sens inverse du délai de propagation.

D2b. Différentes familles :

Actuellement certaines familles sont abandonnées, au profit d'autres plus performantes et moins coûteuses. Nous ne les mentionnerons que pour montrer l'évolution des circuits intégrés.

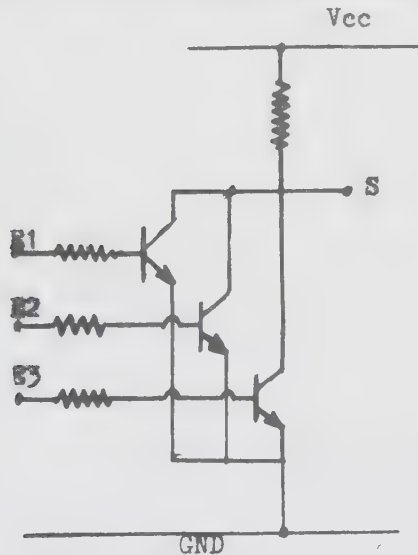
DCTL : La sortie d'une porte est directement liée à l'entrée de la suivante.



Inconvénients : Le fonctionnement dépend de l'équilibrage entre les transistors. En effet, si la tension base-emetteur de l'un des transistors est légèrement inférieure aux autres, il y a phénomène d'accaparement du courant par ce transistor.

Pour que le courant de base dépende moins des caractéristiques d'entrée, on ajoute une résistance en série dans les bases des transistors d'entrée ; d'où la logique RTL.

RTL : Les résistances réduisent l'effet d'accapare-
ment du courant et permettent une plus grande sortance pour l'étage précédent.



Marge de bruit faible : 0,5V entre le '0' et

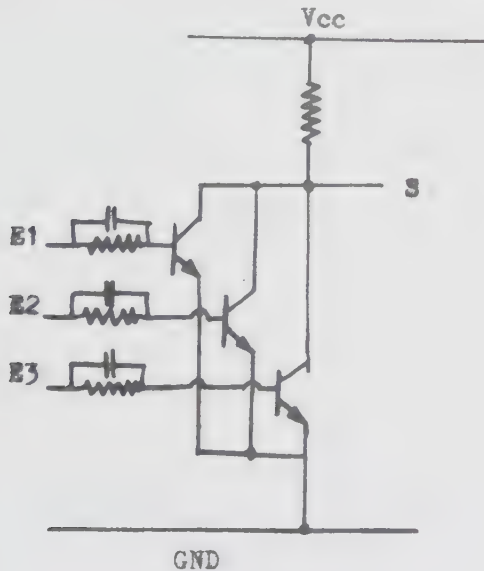
la tension de seuil.

: 0,2V entre le '1' lo-

gique et la tension de seuil.

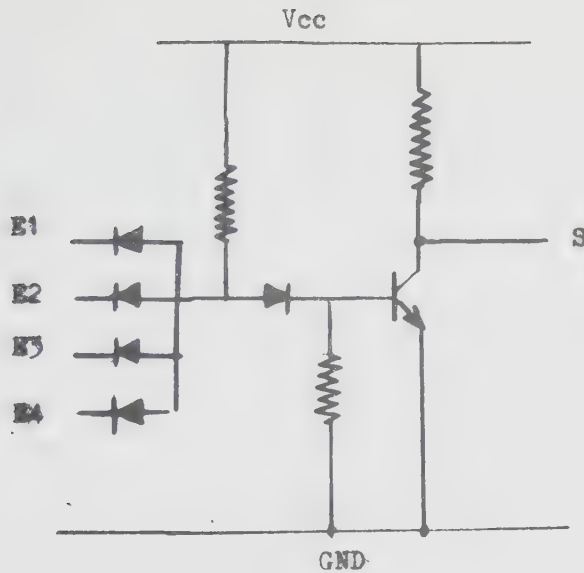
Cependant plus la résistance est grande, plus le temps de commutation est long. Pour augmenter la rapidité du circuit, on ajoutera une capacité en parallèle avec la résistance de base ; d'où la logique RCTL.

RCTL : La logique RCTL représente un bon compromis entre sortance élevée et temps de commutation faible. Cependant les éléments passifs en circuits intégrés ont une surface très importante ; ils sont donc coûteux.



Avec l'étude des familles DCTL, RTL, RCTL, nous voyons que le couplage par éléments passifs est une solution qui ne présente pas de performances élevées et qui est coûteuse. D'où la naissance d'un second type de famille où les éléments de couplage sont des éléments actifs.

DTL : Dans cette famille de circuits, les fonctions logiques sont assurées par des diodes. Une diode série assure le couplage entre les diodes d'entrée et le premier transistor.



La tension de seuil du circuit est égale à la tension aux bornes de la diode série. Elle atteint 1,4V.

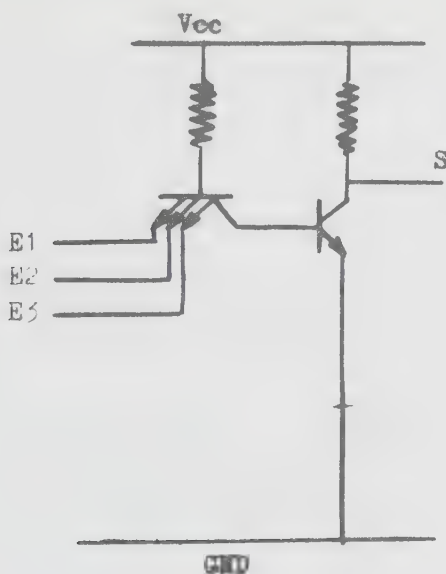
Le niveau logique '0' correspond à la saturation du premier transistor, c'est à dire est faible : 0,2V.

La marge de bruit à l'état '0' est donc 1,2V.

L'entrée au niveau logique '1' correspond à une grande impédance (la diode est contre polarisée). Le niveau logique '1' peut donc être égal à $V_{cc} = 5V$ et la sortance atteint 8.

Les diodes polarisées directement ayant une faible résistance, le temps de commutation est faible et le retard du circuit est de l'ordre de 25 ns.

TTL : Les diodes d'entrée et série de la famille DTL sont remplacées par un transistor multiemetteur. Les diodes emetteur-base remplacent les diodes d'entrée de la logique précédente. La diode base-collecteur est équivalente à la diode série.



La surface du transistor multiemetteur est reduite par rapport à celle de plusieurs diodes.

La tension de seuil est de 1,4V. La marge de bruit à l'état '0' est de 1V.

La vitesse de fonctionnement due à l'étage de sortie est plus grande que dans le cas de la logique DTL. De plus, ce dernier permet une sortance plus élevée.

En conclusion, la logique DTL et surtout la logique TTL semblent bien adaptées pour la construction d'éléments logiques courants.

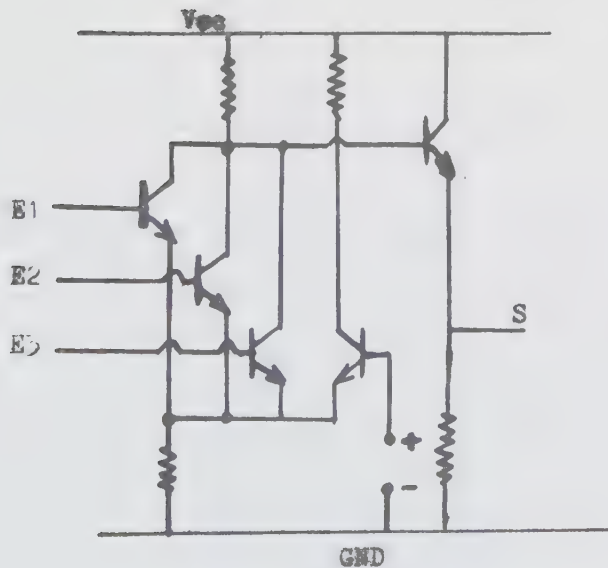
Applications particulières :

Pour quelques applications particulières, il existe quelques familles logiques adaptées :

1. Recherche d'un faible temps de commutation

ECL :

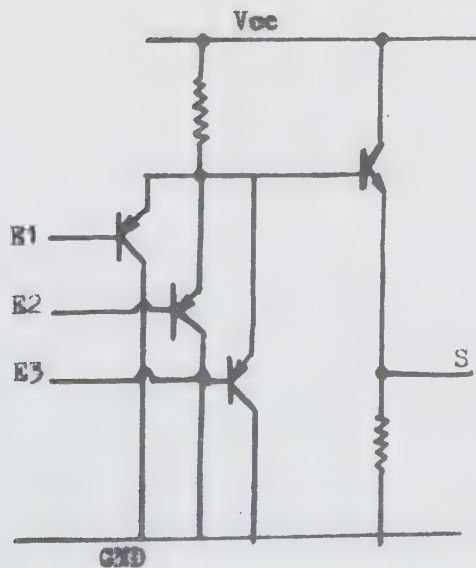
- Les emetteurs des transistors sont couplés et la résistance commune est suffisamment élevée pour servir de source à courant constant.
- Les transistors de commutation ne sont plus saturés, d'où gain de temps.



- La tension de seuil est bien définie alors que les niveaux logiques peuvent varier pour un même état.
- La consommation par porte est élevée, de l'ordre de 50mW.
- L'étage de sortie a une faible impédance ; la sortance est élevée.

CTL :

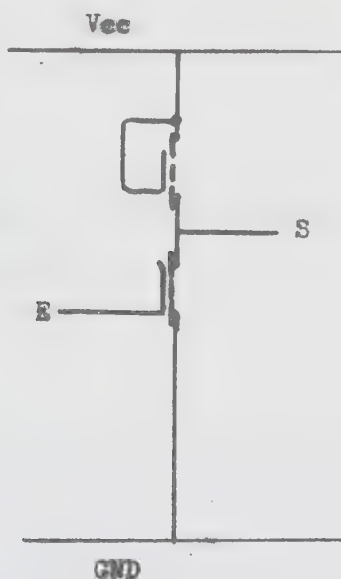
- Les transistors de commutation ne sont pas saturés.
- Faible impédance de sortie ; sortance élevée.
- Forte impédance facilitant les couplages.



En conclusion, les logiques non saturées sont rapides et ont une sortance élevée. Elles ont aussi une grande consommation et une grande marge de bruit.

2. Recherche d'une économie, les temps de propagation n'ayant que peu d'importance:

MOS :



- La logique de commutation est construite à partir de transistors MOS.
- La tension d'alimentation est - 15V.
- Le niveau logique 1 correspond à -11V.
- Le niveau logique 0 correspond à -0,5V.

- La tension de seuil est de $-4V$, donc la marge de bruit est $3,5V$.
- Le temps de commutation est de l'ordre de 250 ns à $1\text{ }\mu\text{s}$.
- L'impédance d'entrée du circuit étant très grande Ce dernier est sensible aux parasites de faible énergie.
- La logique MOS est économique, mais lente.

Nous pouvons dresser un tableau comparatif des caractéristiques des différentes logiques :

	DCTL	RTL	RCTL	DTL	TTL	ECL	CTL	MOS
fonction logique	NON OU	NON OU	NON OU	NON ET	NON ET	OU NON OU	ET OU	NON OU
prix relatif par porte	faible	moyen	moyen élevé	moyen	moyen	élevé	élevé	très faible
temps de propagation par porte (ns)	15	50	30	25	10	5	5	250
consommation par porte (mW)	10	10	10	15	20	50	50	1
marge de bruit moyenne (V)	0,2	0,2	0,2	0,7	1	0,4	0,1	2,5
entrance typique	3	3	3	8	8	5	5	10
sortance maximale	3	4	4	8	12	25	25	5

D2c. Evolution des circuits intégrés

En technologie TTL et MOS, nous assistons actuellement à une évolution vers des circuits intégrés de plus en plus complexes.

Les circuits se classent en 3 catégories :

- SSI : "Small Scale Integration"
Nombre de portes par pastille inférieur à 10.
- MSI : "Medium Scale Integration"
nombre de portes par pastille compris entre 10 et 100.
- LSI : "Large Scale Integration"
nombre de portes par pastille supérieur à 100.

A chaque étage correspond :

- Une augmentation de la fiabilité du circuit.
- Une diminution de volume.
- Une diminution de prix dans le cas de grandes séries.

Chaque circuit pose cependant un problème d'optimisation entre la surface de la pastille et la complexité du circuit.

A chaque catégorie correspond un type de technologie :

- SSI, MSI standards :

Les connections entre chaque élément du circuit se font en une seule étape.

- MSI plus complexes :

Les connections entre les éléments du circuit se font sur deux niveaux. Pour certains circuits complexes, le deuxième niveau peut être déposé sur mesure, après étude.

- MSI hybrides :

Le circuit MSI hybride est un assemblage de pastilles sur support céramique muni d'une grille d'interconnection à film épais. Deux niveaux de connections.

Avec la technologie MOS, l'interconnection des cellules se fait par un ordinateur.

- LSI

Les circuits les moins complexes sont fabriqués par des techniques analogues à celles utilisées pour les circuits MSI.

- LSI hybrides :

Assemblage de circuits MSI sur plaquette de céramique.

- LSI sur plaquette entière :

Sur la plaquette sont reproduits différents circuits élémentaires. Après test, les bons circuits sont reliés entre eux pour réaliser la fonction demandée. Chaque plaquette présente un réseau d'interconnection (3 couches en général) différent. Il est donc nécessaire d'utiliser un ordinateur.

En technologie TTL bipolaire, les mémoires actives et les registres à décalages présentent un réseau simple d'interconnexions. Il est donc probable que ces circuits seront développés.

L'intégration sur plaquette entière en technologie MOS pose des problèmes de connection dus au grand nombre d'éléments et à leur répartition aléatoire. Des mémoires à grande capacité ont été réalisées dans cette technologie.

D2d) Présentation des circuits :

Actuellement, il existe 3 formes standardisées de circuits :

<u>Flat Pack</u>	14 broches
<u>Dual in Line</u>	14, 16, 24...broches
	boîtier céramique (J)
	boîtier plastique (N)
<u>Boîtier métallique</u>	type T05

La forme la plus pratique à utiliser pour réaliser des prototypes est la forme Dual in Line 14 ou 16 broches.

D3. Conclusion :

D'après l'étude du paragraphe précédent, nous choisirons pour construire un opérateur courant, la logique TTL.

L'étude des catalogues de divers constructeurs fait apparaître deux séries TTL privilégiées :

- série 5400/7400 (complétée par les séries 54H/74H rapides et 54L/74L à faible consommation)
- série 9300

Ces deux séries présentent les avantages suivants :

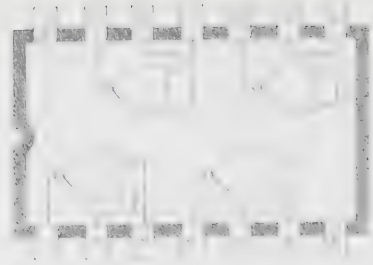
- bonne qualité électrique (Fan Out de 7 à 10)
niveau logique 0 entrée 0 , 0,4V
niveau logique 1 entrée 2,4V
 - performances satisfaisantes (retard de 13ns en moyenne par porte)
 - catalogue complet, en constante évolution vers les circuits complexes à haute intégration (MSI/LSI)
 - fabrication en grande série
- ce dernier facteur est très important car il entraîne 2 avantages :
- facilité d'approvisionnement
 - prix de revient plus faible

Pour des opérateurs particuliers, par exemple les opérateurs contenant des éléments d'Entrée/Sortie, une étude en fonction du problème posé pourra être faite (Par exemple, il sera peut être nécessaire d'utiliser une série de famille DTL compatible avec la logique TTL utilisée pour les opérateurs, dans le cas où l'environnement nécessite une plus grande marge de bruit pour les circuits d'entrée Sortie).

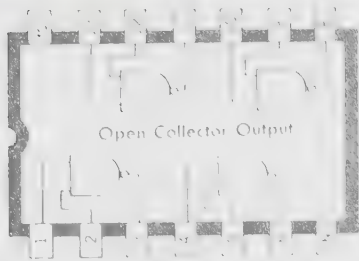
o-o-o-o-o-o-o-o



5400 7400 Quad 2-Input NAND Gate



5402 7402 Quad 2-Input NOR Gate



5403/7403 Quad 2-Input NAND Gate



5404 7404 Hex Inverter



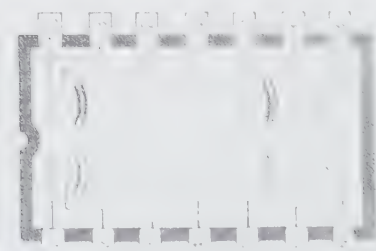
5410 7410 Triple 3-Input NAND Gate



5420 7420 Dual 4-Input NAND Gate



5430 7430 Single 8-Input NAND Gate



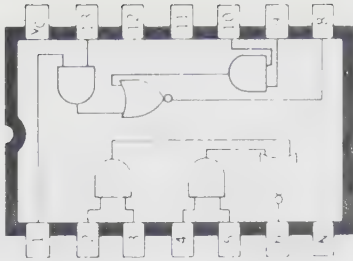
5486 7486 Quad 2-Input OR Gate

E. REALISATION D'UNE STRUCTURE ULM

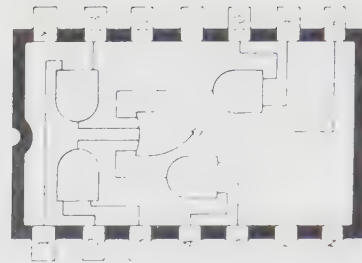
E1. Catalogue des circuits intégrés (Fig. E1)

Les circuits intégrés utilisés pour construire les opérateurs et les ensembles appartiennent à la série TTL 7400.

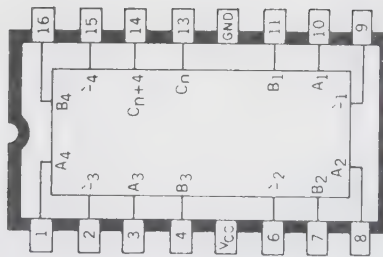
	N°	FONCTION	TEMPS DE RETARD MAXIMUN
<u>Portes</u>	7400	4 NAND 2 entrées	22 ns
	7402	4 NOR 2 entrées	22 ns
	7403	4 NAND 2 entrées "open collector"	22 ns
	7404	6 inverseurs	22 ns
	7410	3 NAND 3 entrées	22 ns
	7420	2 NAND 4 entrées	22 ns
	4931	2 NAND 5 entrées	22 ns
<u>Ensembles</u>	7451	2 AND/OR/INVERT 2 entrées	22 ns
	7454	1 AND/OR/INVERT 4 entrées	22 ns
	7486	4 EXCLUSIVE OR 2 entrées	22 ns
	7483	Additionneur parallèle 4 bits	60 ns
	7475	2x2 Latches	30 à 40 ns
	7474	2 bascules type D	30 à 40 ns
	74170	4x4 Latches	



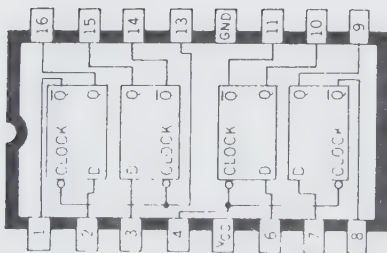
5451/7451 Dual 2-Wide, 2-Input AND-OR-INVERT Gate



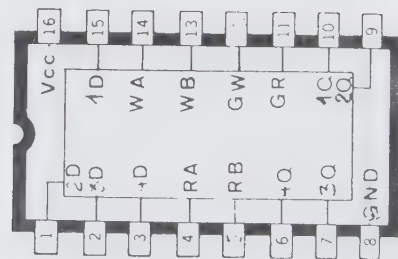
5454/7454 4-Wide, 2-Input AND-OR-INVERT Gate



5483/7483 4-Bit Binary Full Adders



5475/7475 Quadruple Bistable Latches



54170 74170 4-by-4-Register File

E2. Etude hardware des différents opérateurs et ensembles

E2a) Matrice de microprogrammation

E2a1. Ensemble MM: (Fig. E2)

Une microinstruction correspond à une ligne de la matrice de microprogrammation.

Les colonnes sont groupées par champ.

Chaque ligne est couplée à une colonne de chaque champ, par l'intermédiaire d'une diode. Afin de diminuer les connections, il est possible d'intégrer l'ensemble diode-transistor de sortie en les remplaçant par un circuit intégré en logique DTL. Cependant une diode de couplage est équivalente à une diode emetteur-base transistor. Un transistor multiemetteur est donc équivalent à une colonne de matrice.

L'ensemble colonne-transistor de sortie est donc équivalent à un élément logique intégré TTL dont la sortie peut commander directement les circuits TTL de la structure.

La matrice de microprogrammation sera matérialisée par un ensemble de circuits intégrés TTL.

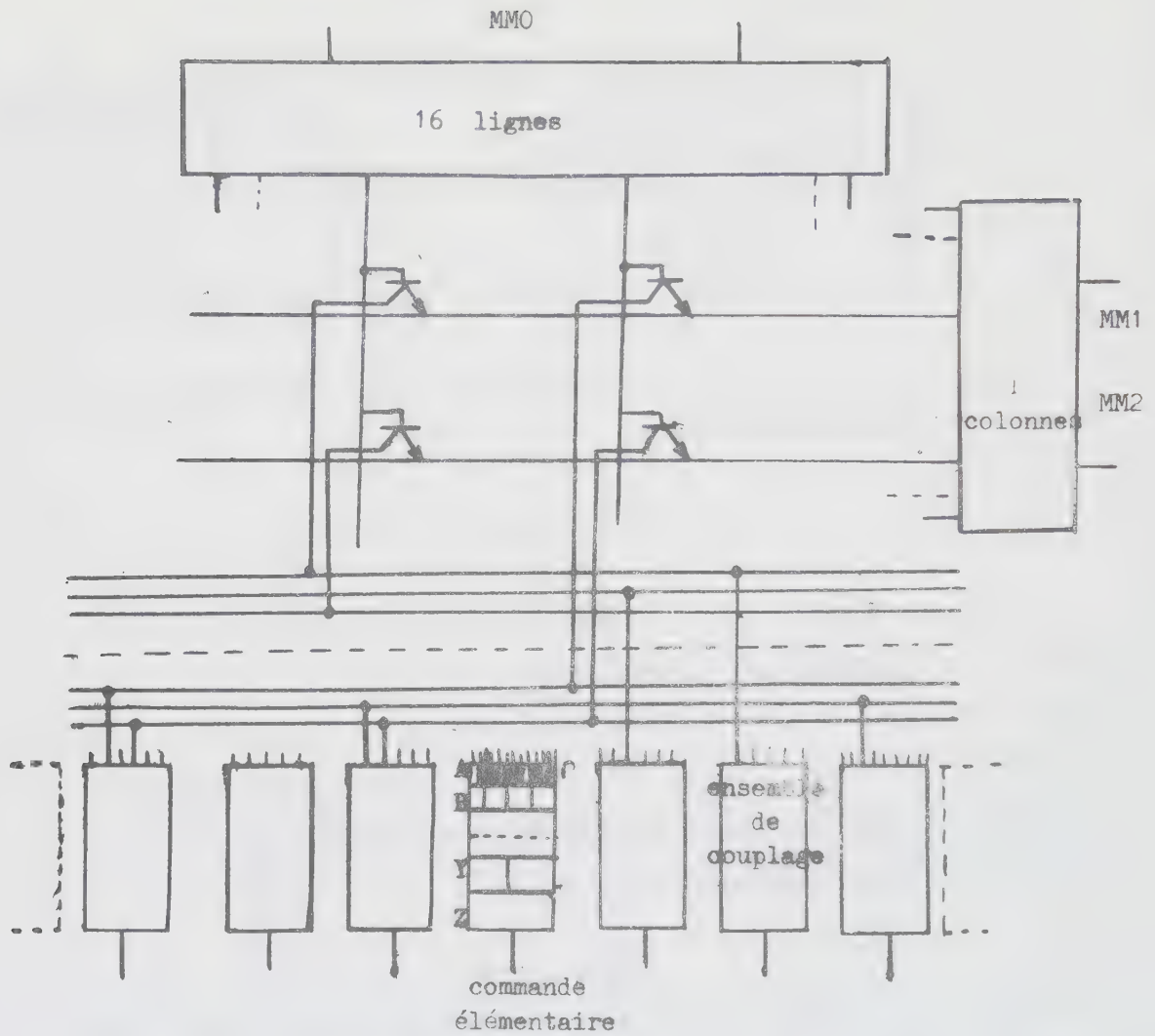


Figure E2a. Organisation de MM

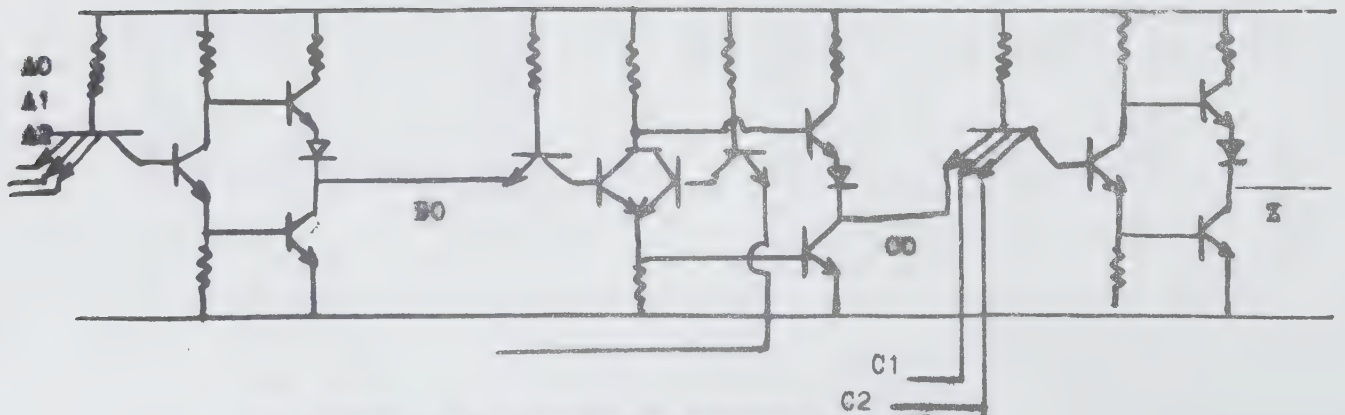


Figure E2b. Détail d'un ensemble de couplage à 3 étages

- Repérage d'un élément de MM :

Chaque circuit de la matrice de microprogrammation est repéré par un numéroteur cinq chiffres définis comme suit :

1. Nombre d'entrée par élément logique
2. 0 à 5 fonction NAND logique positive
6 fonction AND logique positive
8 fonction NOR logique positive
9 fonction OR logique positive
- 3.4. Numéro du circuit dans la série
5. Numéro de l'élément logique sur le circuit.

Ex : 4002-2 : Deuxième NAND à 4 entrées d'un circuit intégré contenant des NAND à 4 entrées (ex : 7420)

Le choix des circuits est défini en fonction des séries qui existent sur le marché, au moment du câblage. Les séries TTL 7400 et 9300 permettent actuellement de construire des matrices constituées de circuits ayant tous 14 broches, donc facilement réalisables sur un support très simple.

Pour le câblage, le numéro de l'élément logique est remplacé par un numéro de broche de deux chiffres.

Ex : 4002 - 06 : Broche 06 du circuit 4002.

- adressage d'un élément de MM :

L'adressage d'une microinstruction se fait par l'intermédiaire d'une matrice de 16 x P transistors NPN. Le nombre P représente le nombre de pages du microprogramme. Il est différent pour chaque application de l'ULM.

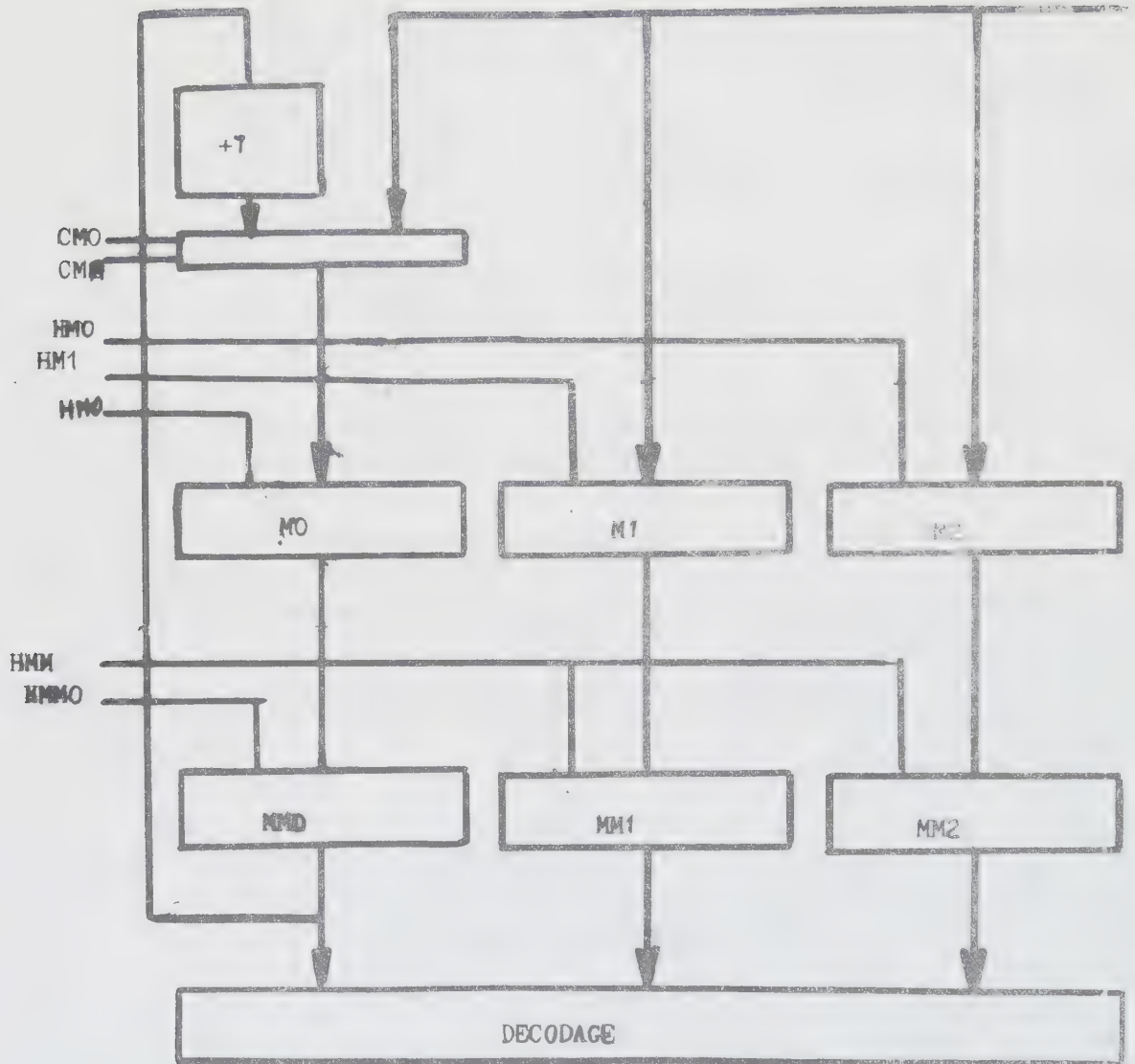


Figure E3a. Organisation de l'opérateur MA

Les bases relatives à une même page sont reliées ensembles.

Les emetteurs relatifs à un même numéro de ligne sont reliés ensembles.

Le collecteur est couplé aux différentes colonnes correspondant à la microinstruction à exécuter.

Un seul transistor, par cycle, est saturé; les autres sont bloqués.

Cette matrice sera construite avec :

- soit des transistors type MPS 3646
- soit les circuits intégrés Dual in Line 16 broches contenant 4 x 2 transistors

Les deux types de matrice ont été testés.
Seul le prix de revient actuel du circuit intégré nous a fait choisir, dans un premier temps, la première solution.

E2a2. Opérateur MA : adressage d'une microinstruction (Fig. E3)

Un même opérateur MA peut adresser 4096 microinstructions.

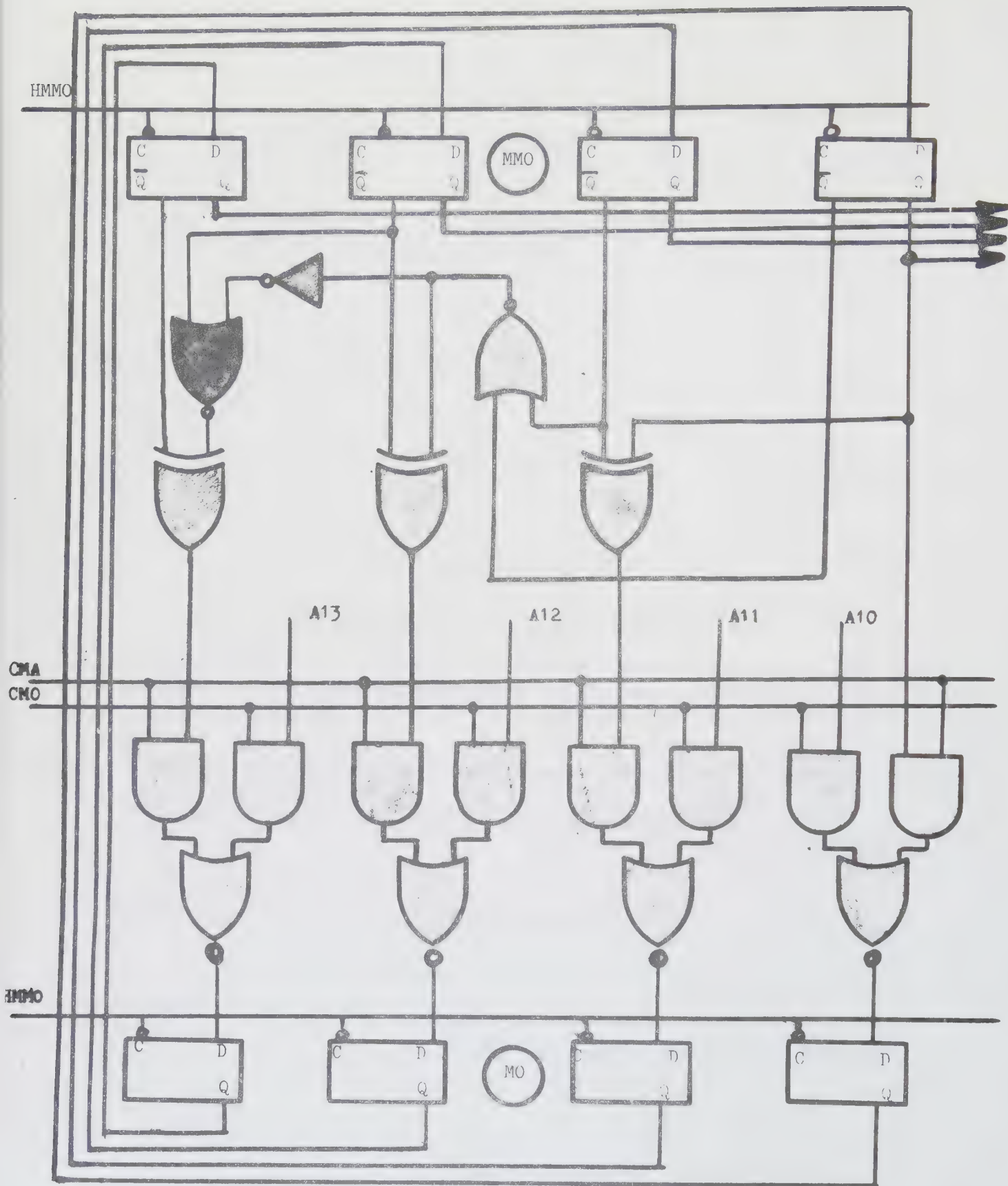


Figure E3b. STRUCTURE DE MO-MMO

- Registre d'adresse :

3 registres de 4 bits : M0, M1, M2.

3 registres de 4 bits : MM0, MM1, MM2.

L'écriture du contenu de A1 dans M1, M2 se fait par commande directe sur les horloges.

L'écriture dans M0 se fait par commande de l'horloge après avoir sélectionné l'information :

soit (MM0) + 1 M0

soit (A1) M0

Sélection : 4 AND/OR/INVERT à 2 entrées.

La recopie de M1, M2 dans MM1, MM2 se fait sur commande lorsque la bascule de saut a été positionnée à 1 à la microinstruction précédente.

La recopie de M0 dans MM0 est automatique à chaque cycle par commande sur l'horloge.

Le +1 se fait par un ensemble comportant : (Fig. E3b)

3 OU exclusif

2 NOR à 2 entrées

1 inverseur

L'initialisation des registres se fait par une commande spéciale. Il est recommandé d'initialiser MM0 à F, valeur très facile à obtenir
Hardware (CMO = CMA = 0)

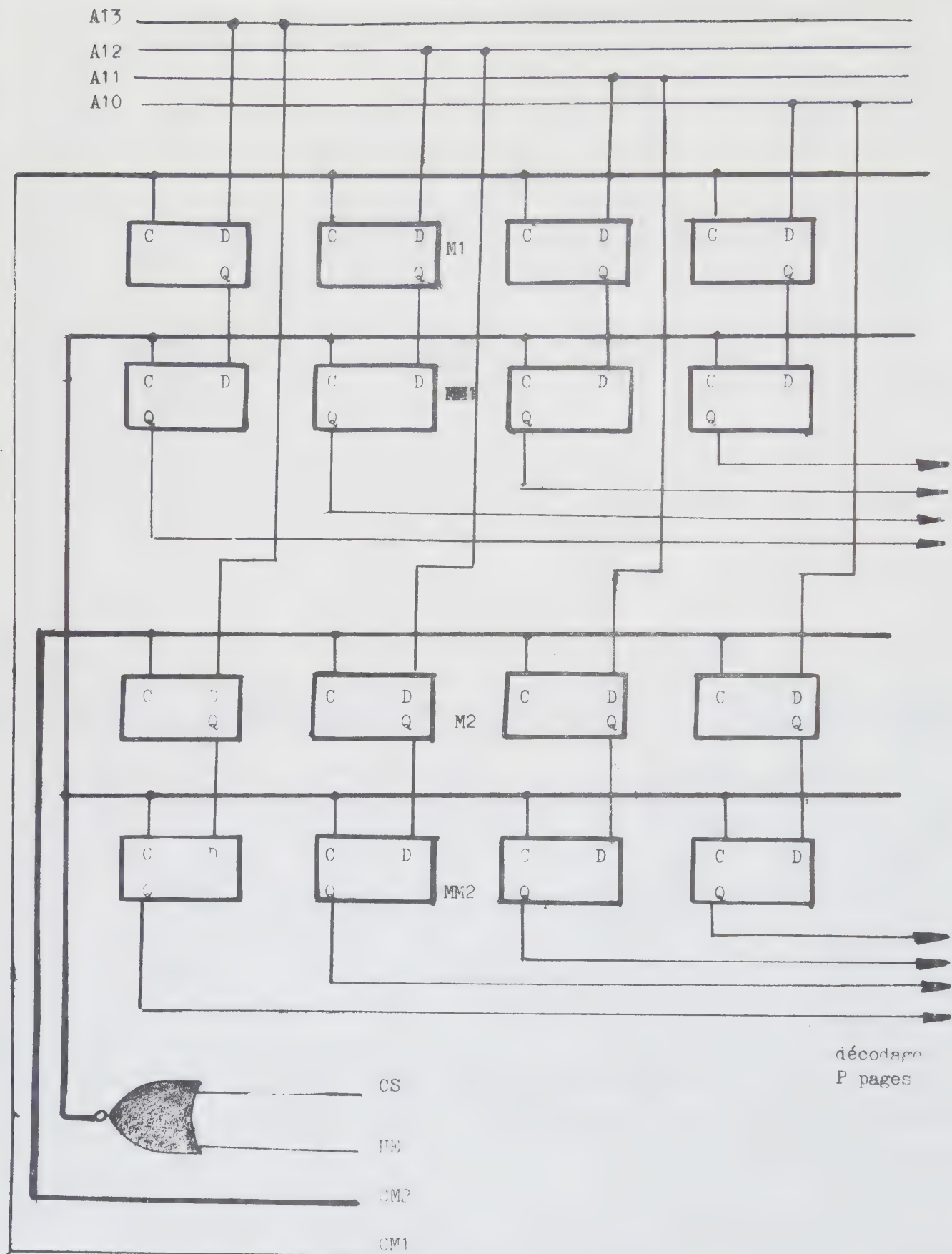


Figure E3c. STRUCTURE DE M1-MM1, M2-MM2.

- Décodage :

Le décodage des registres MM0, MM1, MM2 se fait en 2 parties :

MM0 décodé donne 16 lignes ; chaque ligne commande les bases des transistors d'un même numéro de ligne.

MM1, MM2 décodés donnent jusqu'à 256 lignes représentant les numéros de page. Chaque ligne commande les emetteurs des transistors d'une même page.

E2a3. Ensemble MS : exploitation de la matrice de microprogrammation (Fig. E4)

La logique de sortie est fonction de l'application envisagée. MS est donc un ensemble. Cependant, nous pouvons définir des éléments communs à toutes les applications:

Champ OP (Fig. E4a) :

Chaque colonne représente un état du champ. Elle commande de 1 à n éléments logiques de sortie indépendants, suivant l'état.

Dans le champ OP sont indiquées des opérations telles que saut et gadget qui ne s'effectueront qu'au début du cycle suivant. Il est donc nécessaire de les mémoriser pendant la microinstruction en cours.

La logique de sortie du champ OP contient donc :

- un Flip Flop J de saut
- un Flip Flop G de "gadget" eventuel

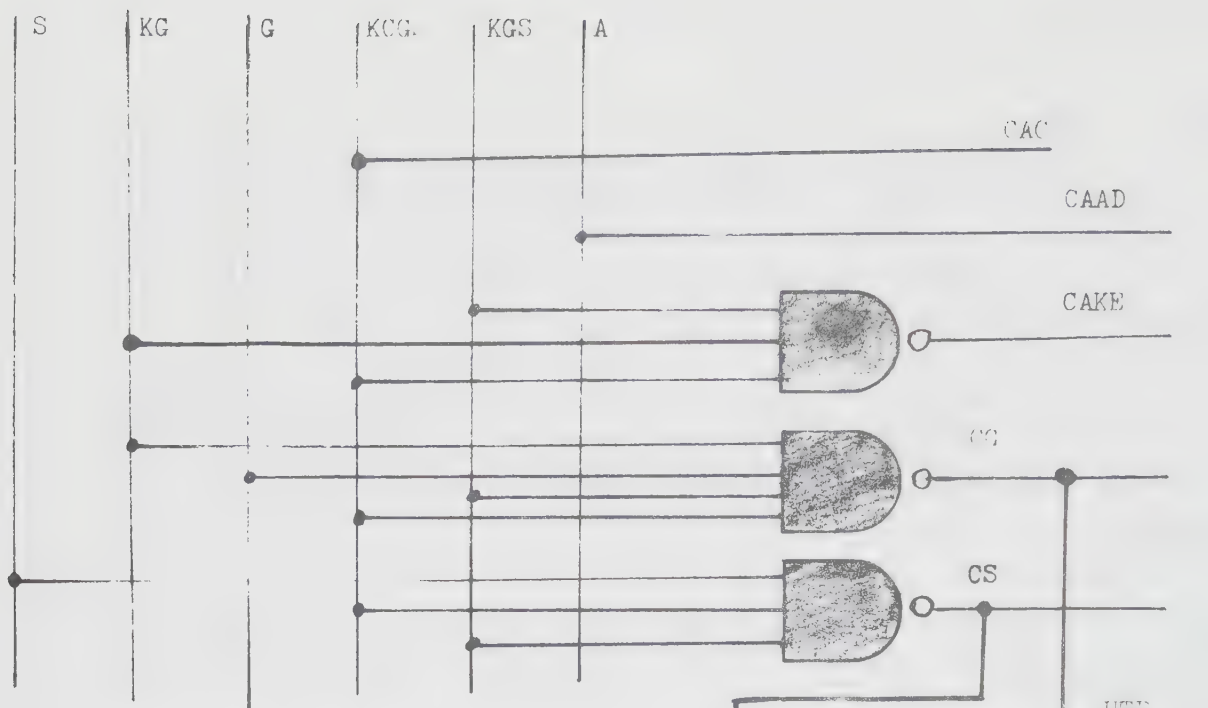
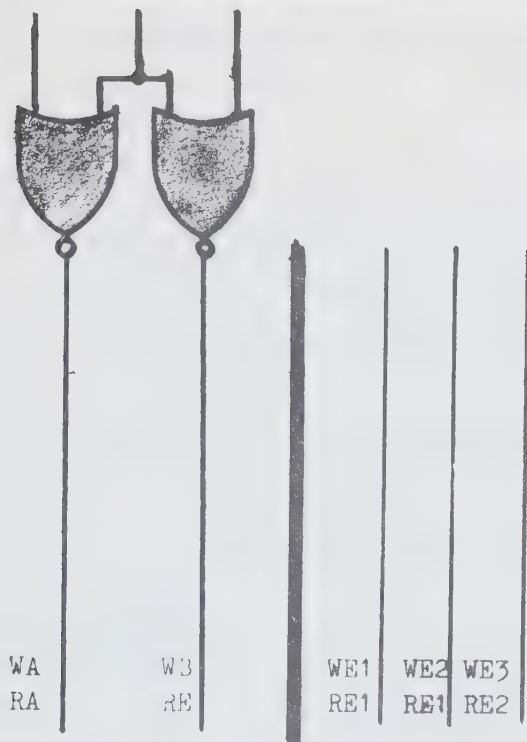


Figure E4a. Champ OP



Demi champ Gauche

Demi champ Droit

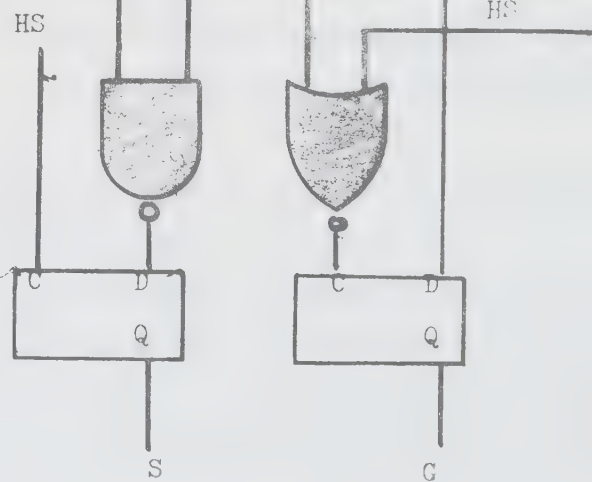


Figure E4b. Champs E-S

Champ 4E - Champ 4S (Fig. E4b)

Pour construire l'Opérateur L, nous avons choisi des circuits intégrés de type 74 170, ensemble de 4 x 4 bascules. Ces circuits, commandables directement par circuits intégrés de la même famille, imposent pour les champs 4E et 4S une structure particulière.

Le choix d'un ensemble parmi 4, de 4 bascules nécessite un adressage de 2 bits. Les adressages en lecture et en écriture sont séparés et peuvent être commandés simultanément. La partie gauche des champs comprendra donc 4 états et sera codée en binaire par 2 NOR à 2 entrées.

La partie droite des champs restera linéaire et commandera :

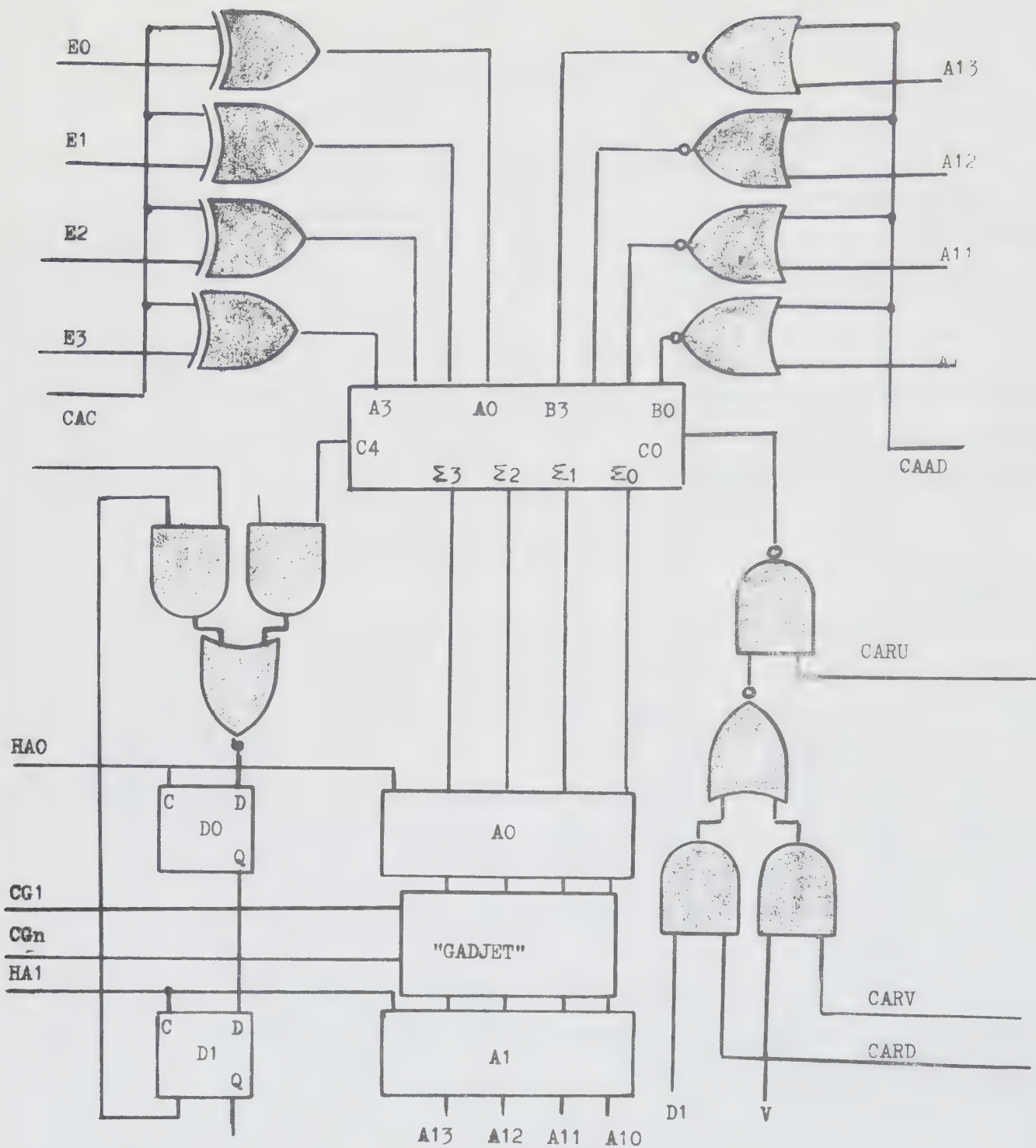
- directement les entrées "read enable" des circuits.
- les entrées "write enable" des circuits à condition que la microinstruction soit exécutable.

L'Ensemble R sera commandé directement par la combinaison des 2 demi-champs de 4E et les 2 demi-champs de 4S.

Champ 1E - Champ 1S (Fig. E4b)

Pour construire l'Opérateur V, nous avons choisi un circuit intégré de type 74170, organisé en 16x1 "latch" indépendants (voir construction de l'Opérateur V).

En organisation 16x1, il n'est pas possible d'écrire et de lire simultanément. Les champs 1E et 1S seront séparés dans le temps.



E5. Opérateur traitement - Boîte à Opérations

Le champ 1E contient la commande Test. Il est prioritaire sur tous les autres champs. La séparation, dans le temps de l'écriture et de la lecture, ne pose pas de problème.

L'opérateur B est commandé directement par la combinaison des demi-champs de 1E et 1S, avec validation pour 1S.

Création d'une constante de 4 bits en champ 1S :

Pour créer une constante de 4 bits en champ 4S, chaque demi-champ donne 2 bits.

E2b. Boîte à opérations (Fig. E5)

L'élément principal de la boîte à opérations est un circuit intégré câblé en additionneur parallèle 4 bits, avec report initial et sortie d'un débordement.

Le résultat de l'additionneur est rangé dans un registre de 4 bits A0 et le débordement dans une bascule D0 par commande directe sur l'horloge .

A0/D0 sont recopiés dans A1/D1 :

- soit directement

- soit par l'intermédiaire d'un élément "gadget" qui peut faire varier l'ordre des bits de A0 ou en remplacer certains ; l'élément gadget sera en général réalisé par des circuits AND/OR/INVERT à 2 ou 4 entrées.

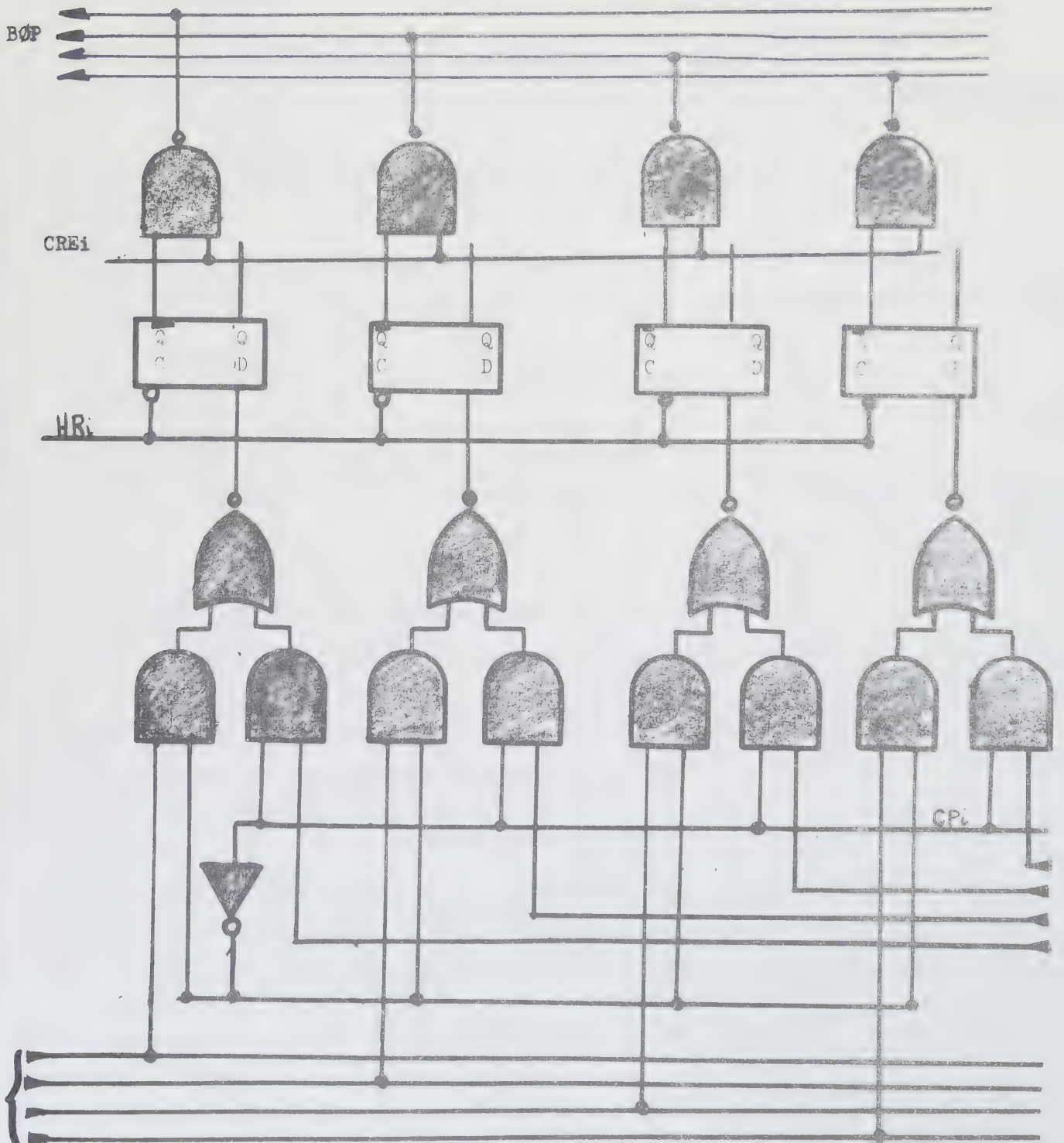


Figure E6a. Organisation de R par latch

L'entrée dans la boîte à opérations se fait par l'intermédiaire de deux bus convergents :

- Bus d'entrée d'information hexadécimal. Il arrive directement sur l'entrée du complémenteur.

- Bus de report initial, 1 bit

Le complémenteur est un ensemble de 4 OU exclusif.

Pour M ou K, en champ OP, le débordement précédent contenu dans D1 est conservé si le débordement généré à la microinstruction en cours d'exécution est nul.

La sortie de la boîte à opérations se fait sur deux bus divergents d'information 4 bits et de débordement 1 bit .

E2c. Ensemble R (Fig. E6)

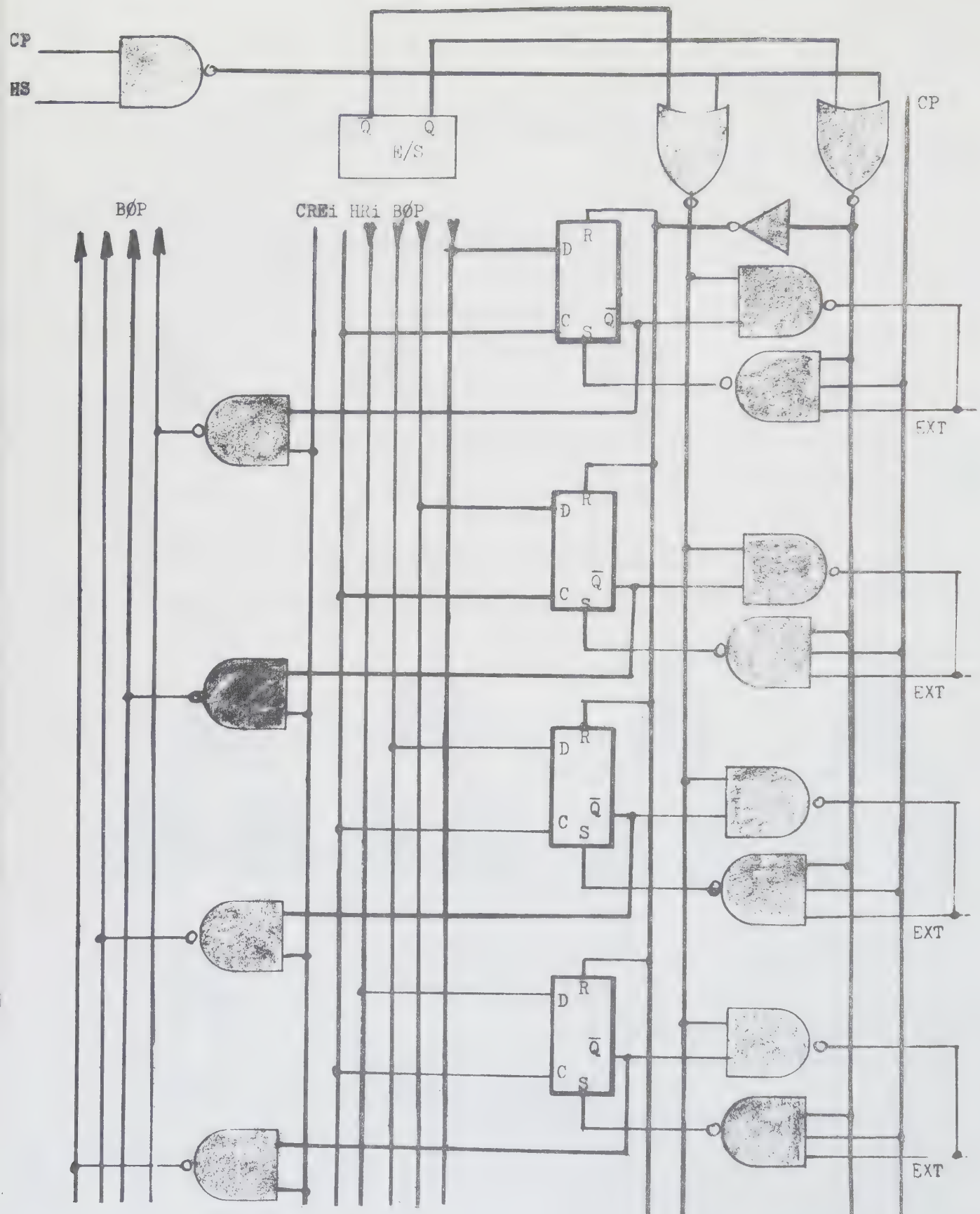
E2c1) Le nombre N de registres réels de l'opérateur R peut varier en fonction des applications.

Les registres Ri, de 4 bits, sont formés :

- soit de 4 "latches"
- soit de 4 bascules set.reset

Les "latches" seront utilisés pour les registres d'entrée ou de sortie. (Fig E6 a)

Les bascules set.reset seront utilisées pour les registres d'entrée et de sortie. (Fig. E6 b)



E2c2) Bus d'écriture

Information :

Pour les registres de dialogue utilisés
comme registres d'entrée nous avons :

- soit (A1) $\longrightarrow$ R1
- soit 4X bits externes $\longrightarrow$ Ri,...R(i+X-1)

L'écriture dans un registre Ri se fait
par action sur l'horloge : (Fig. E6 c)

- soit d'une commande Ri en champ 4S : adressage de
Ri + adressage de l'ensemble R
- soit d'une commande d'entrée parallèle.

La logique de commande comprend
2 NOR à 2 entrées par registre.

Validation :

L'écriture est validée par HTP, si
la microinstruction est exécutable.

La logique de commande comprend donc
1 NOR à 2 entrées de plus.

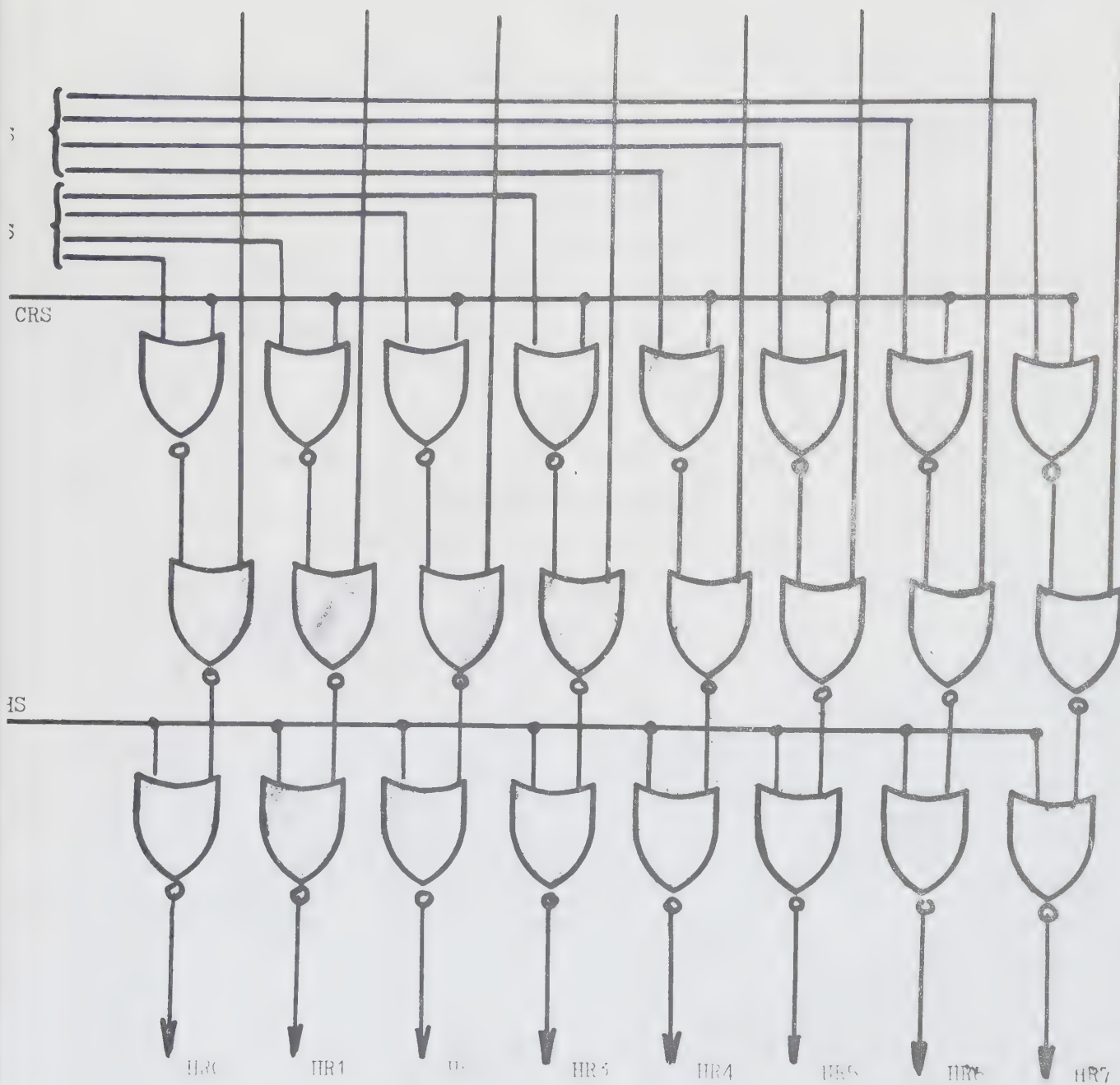


Figure E6c. ADRESSAGE DE R EN ENTREE

E2c3) Bus de lecture

Information :

Lecture - Dialogue : Pour le dialogue avec l'extérieur, les sorties des registres Ri sont directement disponibles, en parallèle.

Lecture - Traitement : Chaque registre R a accès au bus d'entrée de la boîte à opérations à travers une "barrière" de NAND. Le choix d'un registre parmi N est donc une commande de barrière. Pour pouvoir mettre plusieurs portes TTL en parallèle, il suffit de choisir des NAND dont l'étage de sortie est à "collecteur ouvert".

Commande :

La commande se fait directement à partir des valeurs lues dans le champ 4E.

Validation :

Aucune validation n'est nécessaire pour une lecture traitement. La lecture dialogue se fera par ordre de sortie en champ OP.

Extension :

Le nombre de registres R en parallèle n'est pas limité.

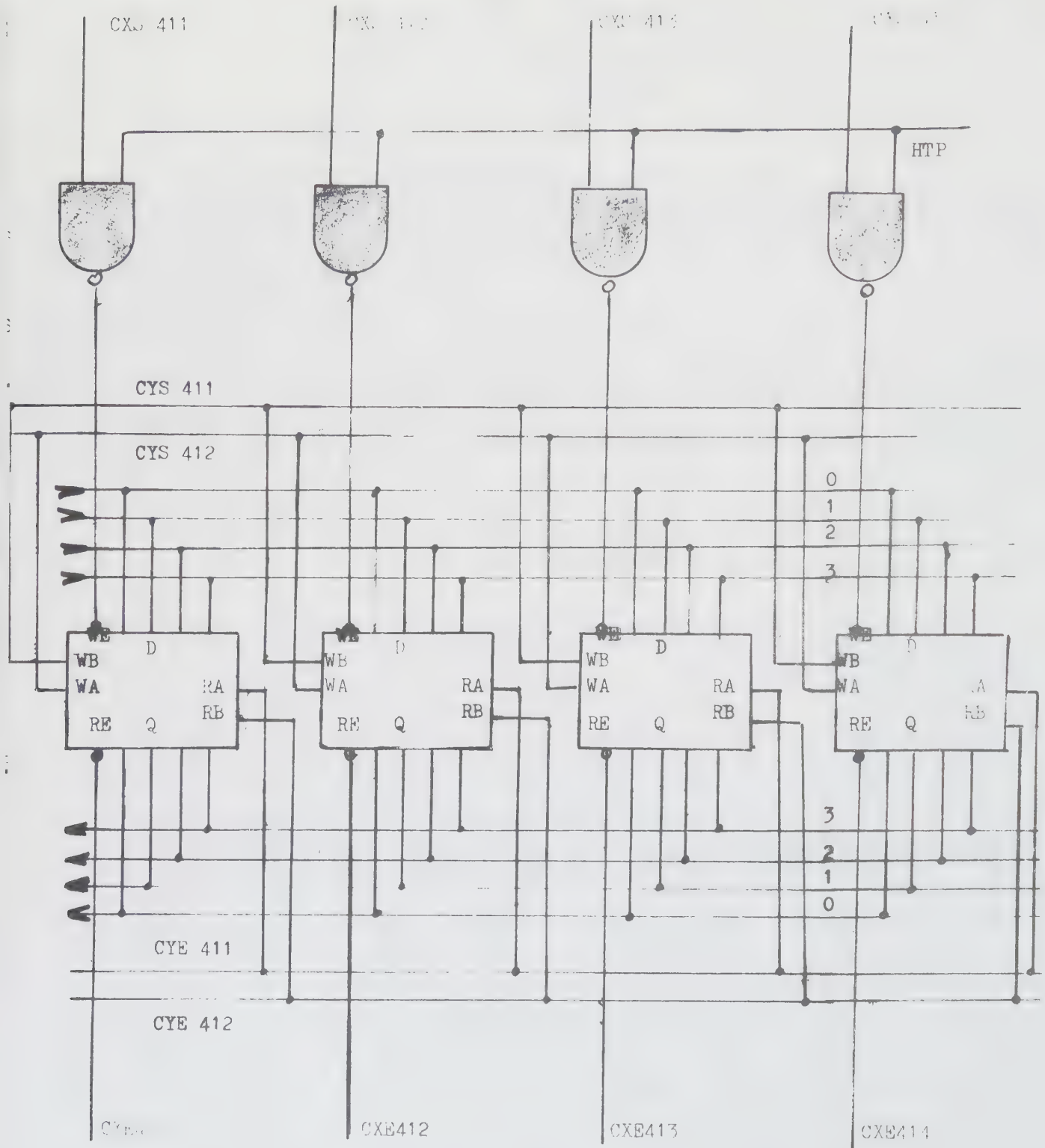


Figure 57. ORGANISATION DE L'OPERATEUR L

E2d. Opérateur L (Fig. E7)

Les 16 registres de 4 bits de l'opérateur L sont constitués par 4 circuits intégrés contenant chacun 4 x 4 bits.

Le circuit 74170 présente un adressage binaire séparé pour la lecture et pour l'écriture. Il est donc possible, pour un même circuit de réaliser simultanément une opération d'écriture et une opération de lecture sur deux ensembles de 4 bits pris parmi les 4 du circuit. Dans le cas où le même ensemble est adressé à la fois en lecture et en écriture, le circuit se comporte comme un registre R.

L'écriture et la lecture d'un ensemble de 4 bits nécessite deux conditions :

- adressage d'un ensemble parmi 4 en écriture ou en lecture
- validation écriture ou validation lecture.

De plus, on peut mettre plusieurs circuits de type 74170, en parallèle. Seule l'information lue dans le circuit validé sera disponible à la sortie de l'opérateur.

La partie gauche des champs 4E, 4S codée en 2 bits, commande directement l'adressage d'un ensemble parmi 4 de tous les circuits intégrés.

La partie droite du champ 4S commande la validation écriture du circuit intégré contenant l'ensemble de 4 bits sélectionné, à condition que la microinstruction en cours soit exécutable.

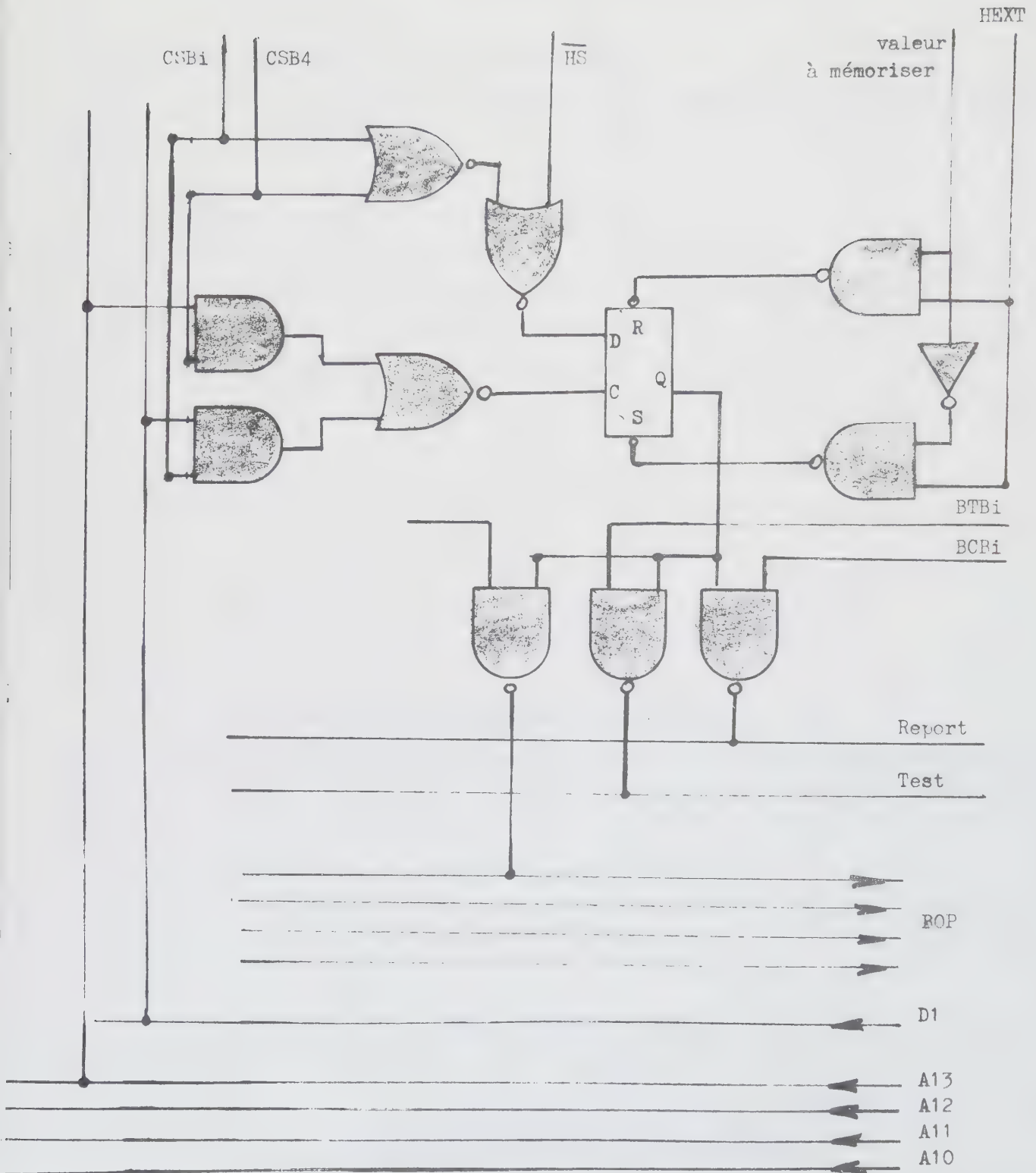


Figure E8. BASCULE B - SET RESET E/S - LATCH ULM

Logique de validation 4 NAND 2 entrées.

Dans le cas d'une utilisation de L en adressage par élément réel de la structure, il faudra ajouter comme circuits logiques :

- choix de l'adressage 2x2 AND/OR/INVERT 2 entrées
- décodage 2 bits 2x4 NAND 2 entrées.

Extension : L'utilisation des circuits 74170 permettrait d'étendre l'opérateur par module de 4 x 4 bits. Cependant l'opérateur complet a été défini pour 16 x 4 bits, et il sera souhaitable d'utiliser des opérateurs complets chaque fois que l'étude de l'application le permettra. L'extension se fait par mise en parallèle de tous les opérateurs L.

E2e) Ensemble B (Fig. E8)

Un ensemble de 4 bascules indépendantes pouvant être traitées soit séparément, soit ensembles, existe en série 900 (9314), mais non encore en série 7400. Il est cependant simple de réaliser des bascules du type latch ou latch set.reset à partir de NAND ou de NOR 2 entrées.

E2e1) Bus écriture :

Information :

Il existe 3 possibilités pour écrire dans les

bascules B :

- soit (A1) B0/B1/B2/B3
- soit (D1) B_i i = 0, 1, 2, 3.
- soit écriture par une ligne externe.

Logique de choix de l'information pour la partie traitement : 4 AND/OR/INVERT 2 entrées.

Commande :

Les commandes sont élaborées à partir des sorties de la matrice de microprogrammation.

Validation :

L'écriture à partir de la boîte à opérations s'exécute si la microinstruction en cours est valide. L'écriture à partir d'une ligne externe est indépendante de la validité d'une microinstruction. Elle dépend de l'état de la bascule.

E2e2) Bus lecture

Information :

Il est possible d'utiliser :

- soit 4 bascules B comme opérande 4 bits
- soit une bascule Bi

Commande :

Commande directe par les sorties de la matrice de microprogrammation.

Validation : aucune validation.

E2f. Opérateur V (Fig. E9abc)

Il existe deux moyens pour réaliser l'opérateur V :

- soit un circuit "scratch pad" organisé 16x1 bits selon la théorie ULM.
- soit un circuit "scratch pad" organisé en 4x4 bits et utilisé en 16x1.

Solution 1 : (Fig. E9a)

L'ensemble des 16 bascules est constitué par un circuit intégré du type 7484. Les bascules sont organisées suivant une matrice 4 x 4.

Chacune d'elles est adressable soit en lecture, soit en écriture.

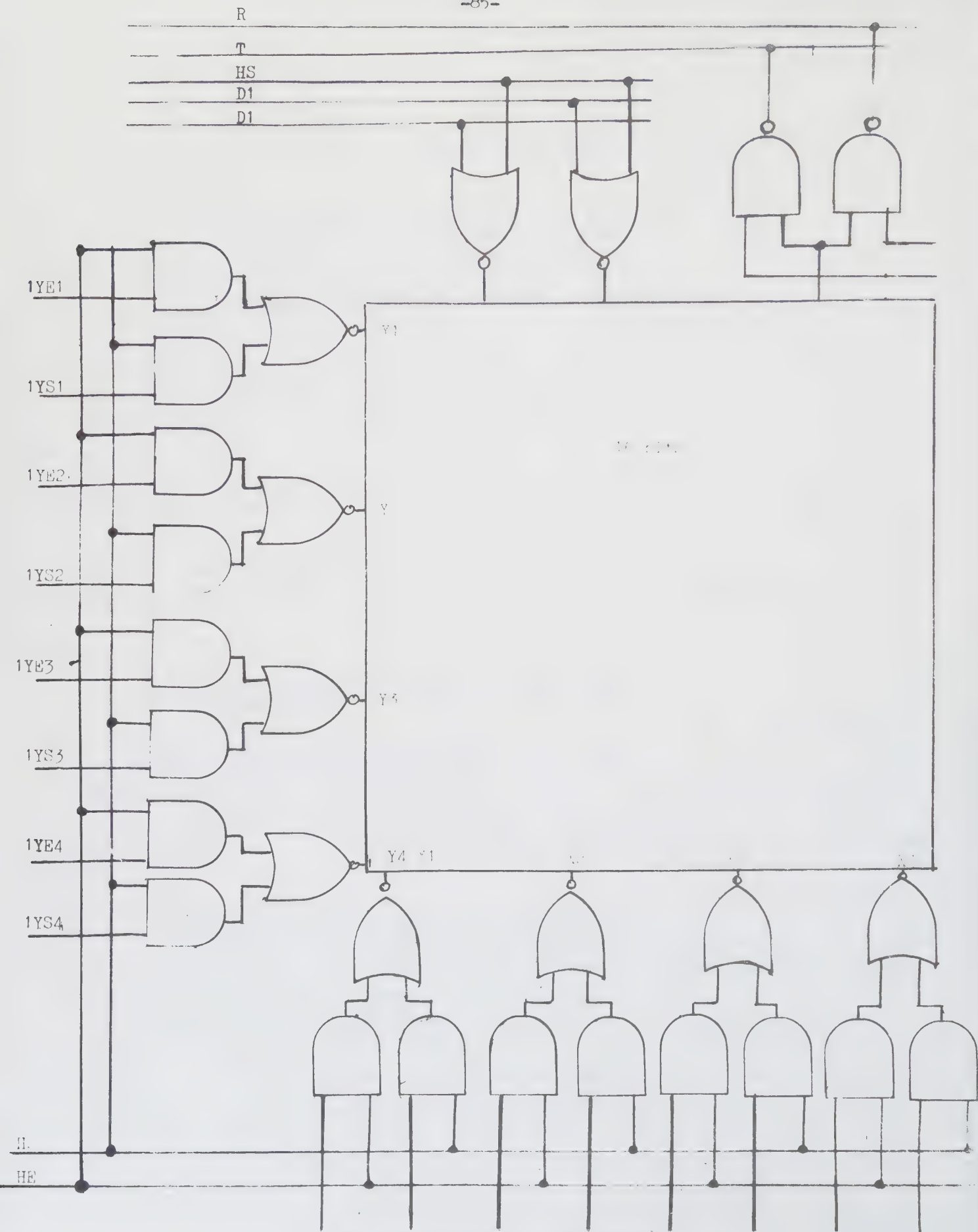
L'adressage se fait par 4 lignes en X et 4 lignes en Y. Pour sélectionner une bascule, il suffit d'avoir une ligne à 1 en X et une ligne à 1 en Y, les autres à 0. Cet adressage est compatible avec les sorties de la matrice de microprogrammation.

Pour une microinstruction, il faut d'abord mettre l'opérateur V en mode lecture, puis en mode écriture.

Ce double adressage, n'imposant aucune contrainte sur le diagramme des temps à l'ULM, nécessite cependant autant de circuits ET/OU/INVERSE à 2 entrées qu'il y a d'entrées d'adresse.

Mémoire 16x1 bit

ET/OU/INVERSE 2 entrées



E9. ORGANISATION DES OPERATEURS V

-Bus écriture

Information :

La seule opération possible est

$$(D1) \longrightarrow V1$$

Pour écrire 1, il faut mettre W1 à 1 WO à 0

Pour écrire 0, il faut mettre W1 à 0 WO à 1

Pour ne pas écrire, il faut mettre W1 à 0 WO à 0

La partie information sera symétrique et comprendra D1 et
D1.

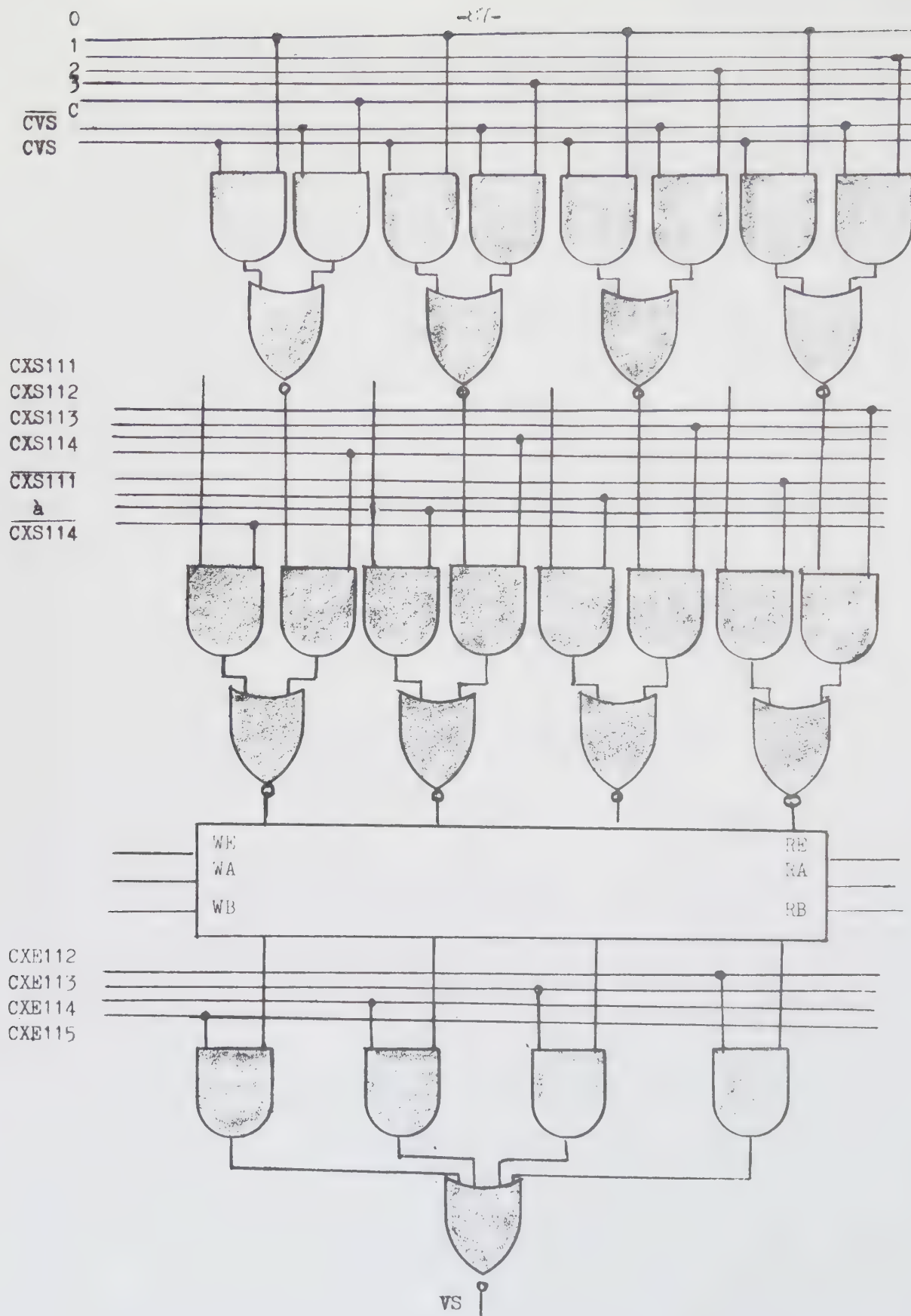
Logique d'écriture : 2 NOR 2 entrées.

Commande :

Commande directe élaborée à partir des sorties de la
matrice de microprogrammation.

Validation :

Validation si la microinstruction est exécutable.



E9b1. Organisation de l'opérateur V - opérateur central -

- Bus lecture

Information :

La valeur de la bascule Vi adressée est mémorisée pendant le cycle de la microinstruction dans un latch V.

L'information est prise directement à la sortie de la bascule V :

- soit pour servir de report initial à l'additionneur.

- soit pour être testée.

Commande

Les commandes de test et de report initial sont élaborées directement à la sortie de la matrice de microprogrammation.

Validation

Aucune validation.

Solution 2 : (Fig. E9b1-2)

L'ensemble des 16 bascules est constitué par un circuit intégré du type 74170, à écriture et lecture simultanées, par groupe de 4 bits.

Remarques :

La structure 4x4 sera conservée si l'entrée comporte un étage supplémentaire de choix entre un ensemble de 4 bits et le débordement contenu dans D1.

Avec seulement deux circuits ET/OU/ INVERSE à 2 entrées il est possible d'écrire dans les bascules V par ensemble de 4. L'information mémorisée est soit :

- Le contenu de D1 dans l'une des quatre bascules
- Le contenu de D1 dans les quatre bascules
- Le contenu de A1 dans les quatre bascules.

Cette remarque entraîne pour la microprogrammation
un certain nombre de possibilités supplémentaires :

- L'écriture du débordement dans 4 bascules V à la fois, peut diminuer la longueur d'un microprogramme d'initialisation

K	-	-	-	3
-	-	RO	-	DSO

initialise : RO à 3

V0, V1, V2, V3 à 0, si pour la microinstruction précédente, le contenu de D1 était 0.

V0, V1, V2, V3 à 1, si le contenu de D1 était 1

- Une utilisation simultanée des champs 4E-1E et 4S-1S
permet de réaliser certaines fonctions particulières :

- Effectuer une opération sur un ensemble de 4 bits par test de l'un des bits de l'ensemble.
- Mémoriser n'importe quel bit de A1 dans la bascule V correspondante :

Bit 1	V0 48 C
Bit 2	V1 59 D
Bit 3	V2 6A E
Bit 4	V3 7B F

- Effectuer une opération logique de masque (ET, logique positive) entre n'importe quel bit de A1 et le contenu de D1 et mémoriser le résultat dans la bascule V correspondante.

Remarque :

Chaque commande d'entrée dans V étant indépendante, il serait possible, en ajoutant des états spéciaux au microprogramme, de travailler sur des éléments d'information de longueur quelconque. Cependant, l'opérateur V présente suffisamment de souplesse pour que la création de telles commandes soit souhaitable.

Comparaison des deux solutions :

Le nombre de circuits de la solution 2 est identique au nombre de circuits de la solution 1, lorsque l'opérateur V est utilisé en configuration 16x1 bits.

Cependant, le circuit 74170 présente des avantages sur le circuit 7484 :

- commandable directement par la sortie de circuits intégrés de la même famille.
- compatible avec un cycle machine de 200 ns.

Il permet en outre de travailler, pour un même opérateur V, soit sur un bit, soit sur des ensembles dont la longueur est celle du mot machine (4 bits).

En conclusion, l'opérateur V sera réalisé par un circuit 4 x 4 bits, autour duquel sera construite la logique de commande permettant d'obtenir :

- soit 16x1
- soit 16x1 et 4x4.

E2g. Organisation des opérateurs de transfert

Chaque opérateur comporte :

- un opérateur central
- un bus d'écriture
- un bus de lecture.

Les bus assurant la liaison entre les différents opérateurs peuvent être définis comme la réunion des bus de chaque opérateur.

E2g1) Bus entrée boîte à opération - 4 bits -

Ce bus convergent d'entrée dans la boîte à opérations est la réunion de quatre bus :

- Bus de lecture de R
- Bus de lecture de L
- Bus de lecture 4 bits de B
- Bus de lecture 4 bits de V

Les éléments logiques de ce bus sont ceux qui composent les quatre bus élémentaires.

E2g2) Bus sortie boîte à opérations - 4 bits -

Le bus divergent de sortie de la boîte à opérations est la réunion de 5 bus :

- Bus d'écriture de R
- Bus d'écriture de L
- Bus d'écriture de 4 bits de B
- Bus d'écriture de 4 bits de V
- Bus d'écriture dans les registres de la matrice de microprogrammation.

E2g3) Bus Report Initial

Information:

Le report initial de l'additionneur de la boîte à opérations est soit le contenu de D1, soit le contenu d'une bascule Vi ou Bi, soit 1; éléments logiques : 1 NAND 2 entrées et 1 AND/OR/INVERT 2 entrées.

Les 4 commandes de report sont élaborées à partir des sorties de la matrice de microprogrammation.

Validation :

Valide quelle que soit la microinstruction.

E2g4) Test

Le bus d'écriture dans la bascule de
test est la réunion de :

- Bus sortie boîte à opérations 1 bit
- Bus lecture 1 bit de V
- Bus lecture 1 bit de B.

E2g5) Bus débordement

Le bus Débordement est la réunion de 2 bus :

- Bus écriture 1 bit de V
- Bus écriture 1 bit de B

E3. Etude des signaux de commande - Diagramme de temps

Notations :

- Toute commande élaborée par la matrice de microprogrammation sera représentée par un ensemble de lettres :

1ère lettre : C

2ème lettre : Opérateur ou Ensemble auquel se rapporte la commande :

A : Boîte à Opérations

B : Ensemble B

L : Opérateur L

R : Ensemble R

V : Opérateur V

X : Partie gauche d'un demi champ
adressage en X

Y : Partie droite d'un demi champ
adressage en Y

3ème, 4ème... lettres :

Identification de la commande.

Si la 2^{ème} lettre est X ou Y, les 3^{ème} et 4^{ème} lettres représentent le champ dans lequel les commandes sont élaborées.

La commande est complétée de 2 chiffres hexadécimaux indiquant le numéro de l'opérateur et le numéro de la commande sur l'opérateur.

Ex : CAAE : commande pour utiliser A1 comme second opérande
(AE en 4E ou M en ØP) .

CXE41A : Commande N°A en X de l'opérateur 1 relatif au champ 4E.

- Toute commande séquentielle sera représentée par un ensemble de lettres :

1^{ère} lettre:H

2, 3^{ème}..lettres : Identification de la commande

Pour une application données de l'ULM, il existe 3 groupes de signaux de commande.

E3a) Signaux élaborés à partir des sorties de la matrice de microprogrammation :

Les signaux élaborés à partir des sorties de la matrice de microprogrammation sont permanents jusqu'à la fin du cycle.

Le tableau suivant donne les états des signaux pour une application de l'ULM où tous les états de chaque champ seraient utilisés par le microprogramme. Pour une application dans laquelle certains états des champs ne seraient pas nécessaires, l'organisation de l'ensemble MS peut varier et l'état des signaux de sortie peut être différent.

CHAMPS	SIGNAL	ETAT SORTIE	ETAT Cde	VAL	COMMENTAIRE
CF	CALF	1	1		
	CAAE	1	1		AE ou AE en C
	CAC	1	1		Complément
	CAAD	1	0		Orlition
	CAGh	1			"Gadget"
	CPh	1			Entrée Sortie
	CHh	1	1	V	Hardware
4E	CARE	1	1		
	CAVE	1	1		
	CABE	1	1		
	CAAE	1	1		AE (AE ou AE en C)
	CLih	1	0		Adressage de L
	CRih	1	0		Adressage de R
	CXE411	1	1		Sélection en lecture
	CXE412	1	1		d'un élément appartenant
	CXE413	1	1		à R ou L.
	CXE414	1	1		Les 2 derniers chiffres
		.	.		sont des chiffres
	CYE411	1	1		hexadécimaux.

CHAMPS	SIGNAL	ETAT SORTIE	ETAT Cde	VAL	COMMENTAIRE
4E (suite)	CYE412	1	1		
	CYE413	1	1		
	CYE414	1	1		
		.	.		
4S	CBS41	1	0	V	
	CLISH	1	0	V	
	CRISH			V	
	CMO	1	0	V	Ecriture dans 10
	CM1	1	0	V	" " M1
	CM2	1	0	V	" " M2
	CXS411	1	1		Sélection en écriture d'un élément appartenant à R ou L.
	CXS412	1	1		
	CXS413	1	1		
	CXS414	1	1		
		1	1		
	CYS411	1	1		
	CYS412	1	1		
	CYS413	1	1		
	CYS414	1	1		
		1	1		
1E	CARD	1	1		RD
	CARU	1	1		RU
	CARV	1	1		RV
	CARB	1	1		RB

CHAMP	SIGNAL	ETAT SORTIE	ETAT Cde	VAL	COMMENTAIRE
1E (suite)	CTV	1	1		
	CTB	1	1		
	CTD	1	1		
	CXE111	1	1		
	CXE112	1	1		
	CXE113	1	1		
	CXE114	1	1		
		.	.		
	CYE111	1	1		
	CYE112	1	1		
	CYE113	1	1		
	CYE114	1	1		
		.	.		
1S	CBS11	1	1	V	
	CVS11	1	1		
	CXS111	1	1		
	CXS112	1	1		
	CXS113	1	1		
	CXS114	1	1		
					
	CYS111	1	1		
	CYS112	1	1		
	CYS113	1	1		
	CYS114	1	1		
					

E3b. Validation CTP

La commande de validation est élaborée à chaque micro-instruction, si CTP = 1. Le signal de validation agit sur les commandes entraînant une modification de l'état de l'ULM.

E3c. Signaux séquentiels

Il existe deux sortes de signaux séquentiels.

E3c1) Les signaux dépendant de la microinstruction en cours :

Les signaux relatifs aux commandes d'écriture dans les registres et les bascules de l'ULM sont validés si HTP = 1. (HS = 1 et CTP = 1)

SIGNAL	NATURE	OPERATION
HAO	0	Ecriture dans DO/AO du résultat obtenu à la sortie de l'additionneur.
HØS	1	Ecriture dans n'importe quel opérateur (4S - 1S) (HS et HTP)
HMM	0	Copie de M1, M2 dans MM1, MM2
HG	1	Ecriture dans la bascule G
HJ	1	Ecriture dans la bascule S

E3c2) Les signaux indépendants de la microinstruction en cours :

SIGNAL	NATURE	OPERATION
HA1	1	Recopie de DO/AO dans A1/D1 avec eventuellement une opération "gadget"
HMO	1	(MO) + 1 MO
HMMO	1	Copie de MO dans MMO

E3d. Diagramme des temps

Le cycle machine comprend deux parties indé-
pendantes (voir microprogramme)

1. La partie adressage de la microinstruction suivante, décodage et obtention des signaux de commande et de validation.
Durant cette période, toutes les opérations indépendantes de la microinstruction adressée sont effectuées, ainsi que toutes les commandes n'entraînant aucune modification de l'état des éléments de stockage de la machine.
2. La partie exécution de la microinstruction, pendant laquelle l'état des éléments de stockage de la machine sera changé si
HTP = 1

Le cycle sera divisé en deux temps :

1. Temps 1 : HE

Copie MO $\longrightarrow$ MM0

M1, M2 $\longrightarrow$ MM1, MM2 éventuellement

A0 $\longrightarrow$ A1

2. Temps 2 : HS

MM0+1 ou A1 $\longrightarrow$ MO

A1 $\longrightarrow$ registre de la structure

Les temps 2 et 1 peuvent être consécutifs.

Les temps 1 et 2 peuvent être consécutifs, si le
front de montée du temps 2 intervient après le front de descente du temps 1

L'addition des 2 opérandes débute au temps 1 et le résultat est pris en compte au temps 2. L'adressage de la microinstruction suivante et l'addition de deux opérandes est, en valeur relative, trois fois plus important que le temps de copie de A1 dans n'importe quel registre de la structure.

Le cycle de l'ULM sera divisé en 4 temps élémentaires :

Temps 1 - 2 : -Adressage de la matrice de microprogrammation.

-Elaboration des commandes correspondantes .

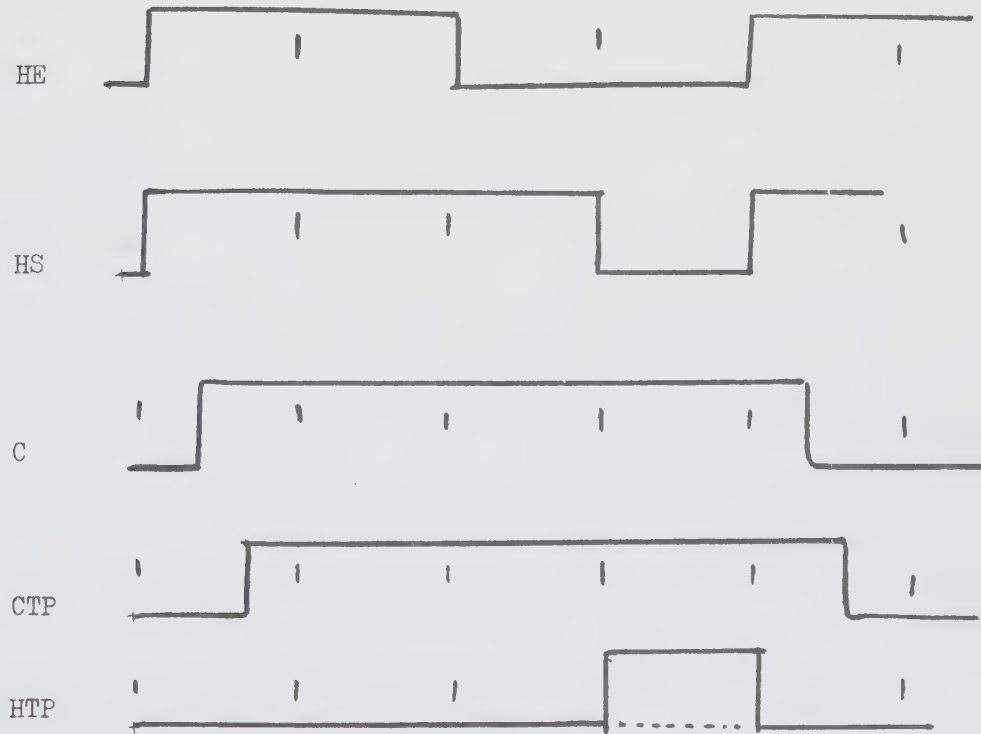
-Exécution de toutes les opérations indépendantes de la microinstruction adressée.

-Recopie de A0 dans A1.

Temps 3 : Exécution de toutes les opérations indépendantes de la microinstruction adressée.

Temps 4 : -Copie du résultat de l'opération dans AO.
-Chargement de MO.

D'où le diagramme des temps :



Le minimum à respecter pour un temps élémentaire est en logique TTL : 50 ns.

Le cycle d'une ULM est supérieur ou égal à 200 ns.

E4. Réalisation d'une ULM

E4a. Etude des cartes d'éléments logiques

Pour garder un maximum de modularité dans l'assemblage des opérateurs, il est nécessaire pour une structure appliquée, de pouvoir disposer d'une carte ou d'un ensemble fixe de cartes par opérateur.

L'étude précédente a montré que chaque opérateur possède un nombre de circuits parfaitement défini, suivant sa complexité.

Opérateur		Option	Nombre de circuits
OP	L	Adressage par MM	5
Ens	R	Suivant nombre de registres réels	
OP	V	Adressage par MM	6
Ens	B	Suivant nombre de bascules	
OP	MA	1 page	7
		4 pages	10
		16 pages	10
		64 pages	11+décodeurs
			MM1 MM2
		256 pages	11+décodeurs
Ens BOP	MS		MM1 MM2
		Suivant nature du microprogramme	
		Sans C	5
		Avec C	+1

La plupart de ces circuits sont des portes logiques NAND ou NOR à 2 entrées. Ces portes forment des ensembles susceptibles, soit d'être intégrés sur un même circuit, soit de s'intégrer à un circuit existant. Le nombre de circuits utilisés diminuera donc en fonction des nouveaux produits qui apparaitront sur le marché.

Il existe donc deux solutions pour réaliser une ULM :

- Soit utiliser une carte standard pouvant contenir 6 ou 12 circuits intégrés, c'est à dire un opérateur complet, ou une fraction d'opérateur. Toutes les cartes seront réunies dans un même bac standard.

- Soit utiliser une carte pouvant contenir de 40 à 100 circuits et cabler directement les opérateurs logiques. Tous les éléments logiques de l'ULM appartiendront alors à une seule carte, cablée à la demande.

Remarques :

Le nombre de circuits intégrés constituant les ensembles et opérateurs est relativement faible. Pour une application courante, il sera de l'ordre de 50 à 60, souvent moins.

Le nombre de circuits intégrés constituant la matrice de microprogrammation ($MA + MM + MS$) dépend de la longueur et de la nature du microprogramme. Il peut représenter une part importante du nombre de circuits total. Nous verrons, pour l'application Générateur de vecteurs, qu'il représente environ 50 % du nombre total des circuits pour 37 microinstructions.

E4b. Choix des connecteurs

Dans le cas d'un montage de l'ULM par opérateurs séparés, les connecteurs servent d'interfaces entre les différents opérateurs et ensembles logiques.

Pour chaque connecteur, il est possible de définir des ensembles fonctionnels de broches :

- KØB : Broches obligatoires d'alimentation
VCC : +5V
GND : 0V
- KIE, KIS : Broches d'information entrée et sortie de l'opérateur.
- KCM : Broches de commande, reliées à la sortie de la matrice de microprogrammation.
- KV : Broches de validation, dépendant de l'état précédent de la machine, ou du mode dans lequel celle-ci est utilisée.
- KH : Broches de commandes séquentielles ou Horloge.

Opérateur	KØB	KIE	KIS	KCM	KV	KH	Total
OP sans complémenteur	2	5	10	9	1	3	30
OP avec complément	2	5	10	10	1	3	31
OP complet avec gadget	2	5	10	11	2	3	33
L adressage simple	2	4	4	6	1	2	19
L adressage multiple	2	4	4	10	1	2	23
RN	2	4+Next	4+N	N+15	1	2	22 + 5N
N	2	4	4	12	1	1	14
MA 16 pages	2	4	6	4	1	1	18
Normalisation interne	2	5	10	12	2	3	34

Pour chaque opérateur ou ensemble, un certain nombre de broches peuvent être normalisées. Un connecteur de 64 broches sera suffisant pour la plupart des cartes.

L'organisation du connecteur définie à la page suivante est valable pour un opérateur. Pour un ensemble, en particulier R, il sera peut être nécessaire de définir sur les broches libres, soit du côté E, soit du côté S, des broches supplémentaires pour entrer ou sortir de l'information en parallèle.

CARTE			
	COMMENTAIRE		COMMENTAIRE
A1	VCC	B1	VCC
A2	1S	B2	1E
A3	4S4	B3	4E4
A4	4S3	B4	4E3
A5	4S2	B5	4E2
A6	4S1	B6	4E1
A7	1S	B7	1E
A8	4S4	B8	4E4
A9	4S3	B9	4E3
A10	4S2	B10	4E2
A11	4S1	B11	4E1
A12	CTP	B12	HTP
A13	commande écriture	B13	commande lecture
A14	" "	B14	" "
A15	" "	B15	" "
A16	" "	B16	" "
A17	" "	B17	" "
A18	" "	B18	" "
A19	" "	B19	" "
A20	adressage MM écriture	B20	adressage MM lecture
A21	" "	B21	" "
A22	" "	B22	" "
A23	" "	B23	" "
A24	" "	B24	" "
A25	" "	B25	" "
A26	" "	B26	" "
A27	" "	B27	" "
A28	adressage binaire E	B28	adressage binaire S
A29	" "	B29	" "
A30	" "	B30	" "
A31	" "	B31	" "
A32	GND	B32	GND

E5. Conclusion :

A partir d'une même structure de base, il est possible de construire n'importe quelle application par mise en parallèle d'autant d'opérateurs qu'il est nécessaire.

L'interface entre les diverses parties de la structure a été étudié de manière à permettre une réalisation aussi modulaire que la conception elle même.

O-O-O-O-O-O-O

F. ECRITURE D'UN SIMULATEUR ULM

Le simulateur ULM et la définition d'un langage d'utilisation en fonction du monde extérieur font l'objet d'une publication séparée. On trouvera dans ce chapitre, seulement quelques notions générales.

Le nombre et la nature des différents opérateurs constituant une ULM spécialisée, sont essentiellement fonction de l'application pour laquelle la structure a été étudiée. Il est donc nécessaire de posséder un outillage "software" permettant à la fois :

- de simuler les opérateurs élémentaires et les contraintes externes spécifiques au problème posé.
- de construire un prototype de l'unité.

F1. Choix d'un langage de programmation

La simulation d'une ULM fait intervenir un grand nombre d'opérations au niveau élémentaire. Pour travailler sur un même type d'ordinateur, un langage d'Assemblage tel que l'Assembleur 360 ou le COMPASS 3600 est donc intéressant à utiliser.

Cependant, le programme de base est destiné à être complété, au fur et à mesure que les contraintes nouvelles apparaîtront dans l'étude des applications.

En particulier, il faudra pour chaque application simuler un "monde extérieur".

Chaque utilisation peut concerner un usager différent.

Il faudra donc utiliser un langage évolué permettant :

- une mise au point simple et rapide.
- une exploitation sur plusieurs types d'ordinateurs.

En conclusion la programmation devra être faite parallèlement en deux langages :

- le COMPASS 3600 : exploitable directement sur CDC 3600
- le FORTRAN IV : pour les usagers possédant un autre type d'ordinateur et étudiant eux-mêmes leur application ULM.

F2. Organisation du simulateur

(voir brochure Simulateur)

F3. Utilisation du simulateur de microprogramme ULM

F3a. Simulation des opérateurs :

Chaque opérateur est simulé au niveau de ses composants de stockage (Mémoire) et de traitement (Boîte à opérations).

F3b. Simulation niveau microinstructions

F3b1) Analyseur syntaxique :

L'analyseur syntaxique a pour but de vérifier l'écriture du microprogramme. Il détecte les codes invalides.

F3b2) Detecteur d'incompatibilité Hardware :

Chaque opérateur possède des contraintes hardware d'utilisation. Ces contraintes sont simulées dans les opérateurs. Toute erreur doit être détectée avant exécution du microprogramme.

Ex : Contrainte : Impossibilité de lire et d'écrire dans le même registre au cours d'une même microinstruction.

F3b3) Enchaînement des microinstructions :

L'enchaînement des microinstructions, par simulation des registres d'adressage de la matrice de microprogrammation, permet de détecter les erreurs de traduction de l'ordinogramme ULM ou de conception du système.

F3b4) Visualisation :

La visualisation de l'état de l'ULM après exécution d'une microinstruction a pour but la mise au point rapide d'un microprogramme.

Elle peut se faire :

- soit après exécution d'une microinstruction donnée
- soit après chaque microinstruction, en pas à pas
- soit pas à pas entre 2 microinstructions données.

F3c) Simulation niveau système

La simulation d'un "monde extérieur" nécessite une adaptation pour chaque application ; cependant, il est possible de définir 2 types de sous programmes "extérieurs" :

- Entrée Sortie

Entrée Sortie sous contrôle de l'ULM.

Ce sous programme permet de vérifier que les échanges entre l'ULM et le monde extérieur se font correctement.

- Interruptions

Chaque système pose un cas particulier. Si le système possède un mode interruption, celui-ci peut être :

- Aléatoire

-"Semi-aléatoire" : c'est à dire aléatoire pendant une période de temps donné, et n'intervenant pas pendant le reste du temps.

- Fixe à un temps bien défini.

Dans les 3 cas, il faut vérifier que toutes les interruptions sont reconnues et que chaque interruption est traitée correctement.

F3d) Implantation de la matrice de microprogrammation

Le cablage de la matrice de microprogrammation dépend de la longueur et de la nature du microprogramme. Ce choix impose des contraintes de cablage.

Le programme considéré est capable, après avoir choisi les composants de l'opérateur MM, de donner le schéma d'implantation de la matrice de microprogrammation et de l'ULM.

Dans un stade ultérieur, les résultats pourront être donnés sous une forme directement exploitable par une machine à cabler automatique.

O-O-O-O-O-O-O

G. REALISATION D'UNE STRUCTURE ULM

G1. Etude du système

La décision de construire une ULM résulte à la fois :

- de l'étude des opérations du système fréquemment utilisées pour lesquelles la microprogrammation permet de réduire le temps d'exécution.
- de l'étude des opérations trop complexes pour être câblées et facilement réalisables par microprogramme.

Une ULM se justifie dans un système où un certain nombre d'opérations sont microprogrammables et où la microprogrammation donne au système une plus grande souplesse d'utilisation et éventuellement diminue son prix de revient. La décision de choisir une ULM est le résultat d'une étude comparée entre ce qui est microprogrammable et ce qui restera câblé dans le système à réaliser.

Le but des paragraphes suivants est d'étudier les différentes étapes entre la définition d'une ULM et sa réalisation pratique.

G.2 Ecriture du microprogramme

L'écriture d'un microprogramme se déroule en 3 étapes :

G2a) Microprogrammes élémentaires

- Traduction de chaque microprogramme en une suite de micro-instructions, constituant un programme élémentaire.
- Si le microprogramme élémentaire dépasse 16 microinstructions le saut peut s'effectuer :
 - soit dans une page quelconque à définir lors de l'assemblage.
 - soit dans une page dont le numéro est fixé par le microprogramme élémentaire.

Exemple : Un ordinogramme se traduit par un ensemble de 25 microinstructions, dont un boucle sur 16.

Supposons que dans la boucle, il ne reste qu'une seule microinstruction dont le champ 4S soit libre et que la microinstruction précédente ne permet pas d'introduire n'importe quelle constante.

Dans ce cas, M1 ou M2 ne pourront être chargés que d'une valeur fixe déterminée par les contraintes du microprogramme. Le saut s'effectuera dans une page connue.

G2b) Recherche de "noeuds" de microprogrammation

Dans un microprogramme, il existe souvent des ensembles de microinstructions pouvant servir à plusieurs ordinogrammes élémentaires.

Pour chacun de ces ensembles, il faut définir :

- Les points d'entrée.
- L'état de l'ULM, après exécution de la microinstruction précédant le point d'entrée.
- Les points et les conditions de sortie.

Ces ensembles forment des 'noeuds' autour desquels viendront s'articuler les éléments non communs des microprogrammes élémentaires.

G2c) Assemblage des microprogrammes élémentaires

L'assemblage des microprogrammes élémentaires se fait par gestion des registres M0, M1, M2.

Le microprogramme ainsi obtenu est la traduction exacte de l'ordinogramme général établi pour l'application ULM étudiée.

G.3 Etude du microprogramme

Le microprogramme est écrit en fonction d'une application donnée.

G3a) Etude des échanges entre éléments ULM

Une étude approfondie des échanges à l'intérieur de l'ULM peut permettre une meilleure utilisation des possibilités Hardware des opérateurs et des ensembles.

Exemples :

- L'emploi du circuit 9314 (Quad Latch) comme registre réel permet une remise à zéro sans passer par la boîte à opérations.
- L'emploi du circuit 74170 (4x4 latch) permet de diviser l'opérateur L en 4 sous-opérateurs indépendants. L'adressage en lecture ou en écriture peut être simplifié en choisissant bien les affectations des registres L. Il est possible de lire ou d'écrire dans plusieurs registres L situés dans des circuits différents.
- Les bascules de l'opérateur V sont disponibles 4 par 4 à la sortie de l'opérateur. Il est donc possible d'effectuer entre 2 bascules V d'un même ensemble, une opération logique dont le résultat sera équivalent à la lecture d'une bascule.

L'étude des échanges à l'intérieur de l'ULM entrainera la suppression de certains éléments et une nouvelle affectation pour d'autres.

G3b) Etude des particularités du microprogramme

Dans certains cas, il peut s'avérer avantageux de remplacer une suite de microinstructions par un cablage simple. Aucun exemple ne peut être donné à priori. Nous verrons dans le paragraphe H deux applications de cette étude. (Acquisition de 3 caractères, masque de R² en écriture)

G3c) Etude du nombre d'éléments de couplage de la matrice de microprogrammation

Considérons les parties gauches des champs de commande des bus 4E, 4S, 1E, 1S.

Pour commander les opérateurs, un codage des 4 états du champ sur deux bits est nécessaire :

- Etat 1	+	0	0	0	→	1	1
- Etat 2	0	+	0	0	→	0	1
- Etat 3	0	0	+	0	→	0	0
- Etat 4	0	0	0	+	→	1	0

Le résultat du codage serait le même si nous avions attribué 0000 à l'état 1. L'absence d'élément de couplage est considéré comme un état.

Pour économiser des éléments de couplage, il suffira d'affecter aux éléments adressés par 11, les états les plus chargés du microprogramme.

Cette étude entraîne l'écriture d'un nouveau microprogramme,
plus concis, qui tiendra compte :

- des possibilités hardware des ensembles et opérateurs,
- des particularités du microprogramme initial,
- des états du microprogramme initial.

G.4 Etude de la matrice de microprogrammation

L'étude du paragraphe précédent a montré une évolution vers une matrice de microprogrammation plus réduite.

Le microprogramme réel étant écrit, une étude globale de la matrice résultante peut amener de nouvelles simplifications hardware.

Exemples :

- Aucune microinstruction ne contient une commande de lecture et une commande d'écriture dans V.
- Il existe une correspondance d'adressage entre les opérateurs, pour chaque microinstruction où ils sont adressés.

- Aucune microinstruction ne contient à la fois une commande en champ 1E et une commande en champ 1S.

Le nombre d'éléments de couplage de la matrice peut être réduit par de telles remarques.

Cette étude nécessite une connaissance approfondie de l'organisation des champs dans leur ensemble et de chaque champ en particulier.

G.5 Réalisation de l'ULM

G5a) Pour passer du microprogramme à la matrice de microprogrammation, il faut exécuter les opérations suivantes :

1. Choix d'une configuration de champ OP :

Pour choisir la configuration d'un champ, il faut séparer distinctement la notion d'état de MM, de la notion de commande élaborée par MM.

L'état de MM est la propriété, pour une colonne de la matrice de microprogrammation, de passer de 0V à 5V, suivant qu'elle est, ou non, adressée.

La commande de MM est la propriété pour un élément, de déclencher ou non une opération dans la structure.

Un ou plusieurs états de MM entraînent une ou plusieurs commandes élémentaires, sans qu'il existe de relation à priori entre état et commande.

Ex : état KCA (OP) : commandes K, C, A.

état AE (4E) : commande AE

état Loo (4E Droit) : commande Loo

état Loo (4E Gauche) }

Le choix d'une configuration de champ dépend donc de la nature des états de MM et du nombre de commandes élémentaires à élaborer. Pour aider un utilisateur à choisir, il convient de poser certaines définitions :

- coefficient de champ d'un état : Le coefficient de champ

CP d'un état est égal au nombre de commandes élémentaires réelles élaborées à partir de cet état.

Ex : CAS : CP = 3

- coefficient de charge d'une commande élémentaire : Le coefficient de charge CG d'une commande élémentaire est égal au nombre de combinaisons dans lesquelles est incluse la commande élémentaire, pour un même micro-programme

Ex : K, KCA, KP(1), KA : CG(K) = 4

CG(A) = 2

Soient N le nombre d'états différents du champ, Q le nombre de commandes élémentaires à élaborer, U_i le nombre de fois que l'état i est utilisé dans le microprogramme, j l'indice des commandes élémentaires.

Pour une configuration donnée du champ, nous allons calculer le nombre C d'éléments de couplage nécessaires pour obtenir en sortie des commandes élémentaires indépendantes :

Pour un champ à organisation binaire : $C = \sum_{i=1}^N CP_i \cdot U_i$

Pour un champ à organisation non binaire : $C = \sum_{i=1}^N U_i + \sum_{j=1}^Q CG_j$

Dans les deux cas, le nombre de broches de sortie de MS est égal à Q .

Etude de ΔC :

$$\Delta C = \sum_1^N (CP_i - 1) U_i - \sum_1^Q CG_j$$

Tous les états pour lesquels $CP=1$ n'interviennent pas dans la valeur de ΔC .

Posons : $\gamma_i = CP_i - 1$

n_γ = nombre d'états ayant le même coefficient
de champ $\gamma + 1$

m = valeur maximale de γ_i

$$\Delta C = \sum_{i=1}^m n_\gamma \gamma_i U_i - \sum_{j=1}^Q CG_j$$

Posons : $A = \sum_{i=1}^N n_\gamma \gamma_i U_i$

$$B = \sum_{j=1}^Q CG_j$$

A dépend du nombre d'états ayant un coefficient de champ supérieur à un et du nombre de fois que chaque état est utilisé au cours du microprogramme. Il est fonction de la longueur du microprogramme.

B dépend du coefficient de charge de chaque commande élémentaire. Il est indépendant du nombre d'utilisations de chaque état, donc de la longueur du microprogramme.

Si $\Delta C > 0$: Champ non binaire, linéaire.

Si $\Delta C < 0$: Champ binaire

$$\text{Pour } U_i = 1, \quad \sum_{n=1}^N n \gamma_i = \sum_{j=1}^Q C C_j$$

Donc les deux solutions linéaire et binaire sont équivalente, pour $U_i = 1$ quelque soit i .

Si : $\exists i$ tel que $U_i \neq 1$, le facteur A est supérieur au facteur B et ΔC est positif.

Conclusion : Un champ devant élaborer des commandes élémentaires indépendantes, ne peut avoir qu'une structure linéaire.

2. Comptage du nombre U_i d'utilisation de chaque champ.

3. Utilisation du tableau d'équivalence entre le nombre U_i , l'état positif ou négatif de sortie et le ou les circuits de MM.

Nb. de microinstructions	Circuits de couplage	Sortie	Etat
1	10XX	-	+
2	20XX	-	+
3	30XX	-	+
4	40XX	-	+
5	40XX 10XX	28XX	-
	50XX	-	+
6	30XX 30XX	28XX	-
	40XX 20XX	28XX	-
	50XX 10XX	28XX	-
	80XX -	-	+
7	80XX -	-	+
8	80XX -	-	+
9	80XX 10XX	28XX	-
10	80XX 20XX	28XX	-
11	80XX 30XX	28XX	-
12	80XX 40XX	28XX	-
13	80XX 50XX	28XX	-
13-16	80XX 80XX	28XX	-
17-32	3-4 circuits "NAND"	48XX	
	3-4 circuits "NAND"	28XX	
		+28XX	
		+20XX	+
33-48	5-6 circuits "NAND"	circuits "NOR"	-
		"NOR"+"NAND"	+
49-64	7-8 circuits "NAND"	88XX	-
		"NOR"+"NAND"	+
65-128	9-16 circuits "NAND"	"NOR"+"NAND"	+
129-512	17-64 circuits "NAND"	"NOR"+"NAND"	
512		étage de sortie supplémentaire	

Remarque 1 :

Lorsque la sortie est directe, l'état de la commande obtenue, est toujours positif.

Remarque 2 :

Lorsque la sortie nécessite un seul étage d'éléments logiques, l'utilisation de circuits X9XX inverse l'état de la commande obtenue.

Remarque 3 :

Lorsque la sortie nécessite deux étages d'éléments logiques, l'utilisation de circuits X6XX inverse l'état de la commande obtenue.

Remarque 4 :

La combinaison des circuits X0XX - X6XX - X8XX - X9XX permet d'obtenir un état de commande fixé.

4. Affectation des circuits de MM.

5. Utilisation du tableau d'équivalence entre le numéro d'un circuit et l'ensemble de broches d'entrée et de sortie correspondant.

Correspondance entre les éléments logiques et les broches des circuits intégrés.

Série	Broches d'entrée	broche de sortie
800 X	01 - 02 - 03 - 04 - 05 - 06 - 11 - 12	08
500 X - 1	01 - 02 - 03 - 04 - 05	06
500 X - 2	08 - 09 - 10 - 11 - 12	13
400 X - 1	01 - 02 - 04 - 05	06
400 X - 2	09 - 10 - 12 - 13	08
300 X - 1	03 - 04 - 05	06
300 X - 2	09 - 10 - 11	08
300 X - 3	01 - 02 - 13	12
200 X - 1	01 - 02	03
200 X - 2	04 - 05	06
200 X - 3	09 - 10	08
200 X - 4	12 - 13	11
100 X - 1	01	02
100 X - 2	03	04
100 X - 3	05	06
100 X - 4	09	08
100 X - 5	11	10
100 X - 6	13	12

Série	Broches d'entrée	Broche de sortie
280X - 1	02 - 03	01
280X - 2	05 - 06	04
280X - 3	08 - 09	10
280X - 4	11 - 12	13

6. Affectation de chaque broche à une micro-instruction.

G5b) Chaque ULM est destinée à un type d'application bien particulier : unique ou grande série, laboratoire ou commerciale.

Le choix du support d'implantation résulte d'une étude de l'ULM en "situation" :

- Prototype de laboratoire.
- Intégration dans un ensemble existant.
- Element de base d'un ensemble à créer.
- Nombre d'unités à réaliser.

Ces problèmes sont résolus indépendamment de la structure elle-même.

Pour réaliser une première application, les critères définis au paragraphe E sont raisonnables.

G.6 Conclusion

En résumé, les différentes étapes de l'étude et la réalisation d'une ULM sont les suivantes :

1. Etude du système dans lequel s'intégrera l'ULM.
Etude des opérations microprogrammables.
2. Ecriture des microprogrammes élémentaires.
Assemblage du microprogramme.
3. Etude du microprogramme et des échanges à l'intérieur de l'ULM.
Ecriture d'un nouveau microprogramme utilisant au mieux le Hardware ULM.
4. Etude de la matrice de microprogrammation.
5. Implantation de l'ULM en fonction de son utilisation.

L'étude et la réalisation d'une Unité nécessite les connaissances suivantes :

1. Structure ULM, pour définir le problème en fonction de la structure.
2. Microprogramme ULM.
3. Hardware ULM, pour écrire un microprogramme utilisant au mieux les opérateurs et ensembles de la structure.

Remarque :

Le software ULM est défini pour chaque application. Son étude n'est pas indépendante de la structure du microprogramme et du hardware. Il n'est cependant pas possible de définir un software généralisé.

Le but du paragraphe suivant est de montrer comment peut être étudiée et réalisée une structure ULM. Le problème posé est celui de la commande d'un périphérique de visualisation, tracé relatif incrémental, à partir d'un calculateur, en commande directe ou à distance.

O-O-O-O-O-O-O

H. ULM APPLICATION : GÉNÉRATEUR DE VECTEURS

H1. Définition

Un générateur de vecteurs est un algorithme capable d'élaborer des éléments d'information, interprétables par une unité de visualisation comme commandes de déplacement de l'élément marqueur d'un point à un autre, en effectuant ou non un tracé.

Il existe deux méthodes pour se déplacer d'un point à un autre :

- soit déplacement continu.
- soit déplacement incrémental.

De plus, le second point peut être repéré :

- soit en coordonnées absolues.
- soit en coordonnées relatives par rapport au premier point.

Il existera donc plusieurs algorithmes de tracé de vecteurs.

Une unité de visualisation est, en général, réalisée pour un type de déplacement et de système de coordonnées donnés.

Le but de l'étude est de voir s'il est possible d'assembler une ULM, dont le microprogramme contiendrait un algorithme de tracé de vecteurs en relatif incrémental. L'ULM ainsi conçue serait capable de commander toutes les unités de visualisation, peu coûteuses, de la classe "relatif incrémental" tel que table traçante ou écran à mémoire.

H2. Algorithme de tracé de vecteurs en relatif Incrémental

En tracé incrémental, le plan de tracé pourra être remplacé par une grille dont le pas est égal à un incrément.

Le but de l'algorithme est de décomposer un vecteur en une suite de vecteurs élémentaires joignant deux points consécutifs de la grille. Pour passer d'un point de la grille à un autre, il existe huit directions possibles :

- parallèlement aux axes
- parallèlement aux bissectrices des axes.

Remarque 1 : Pour pouvoir tracer un vecteur dans n'importe quelle direction du plan, il suffit de savoir le tracer dans l'octant défini par l'axe OX et la première bissectrice, puis de le ramener dans son octant d'origine par produit de 3 symétries maximum (première bissectrice, axe Y, axe X).

Remarque 2 : Dans un octant, il n'existe que deux directions de déplacement utiles, celles parallèles aux droites définissant l'octant.

Remarque 3 : Dans un octant, la commande de déplacement parallèle à un axe est équivalente à la commande de déplacement parallèle à une bissectrice, masquée à droite ou à gauche suivant l'octant.

Conclusion : Pour tracer n'importe quel vecteur, il suffit de connaître :

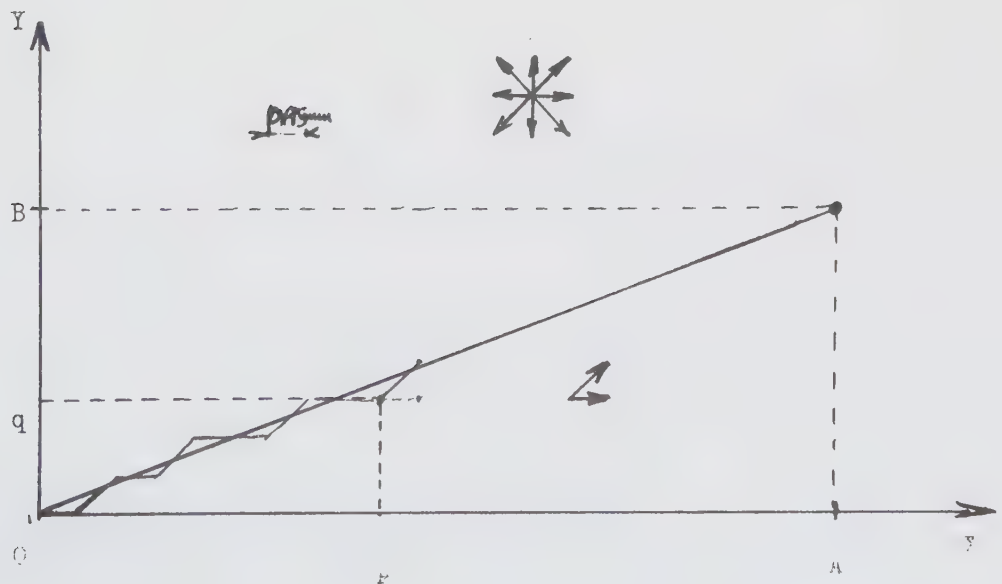
1. Le numéro d'octant dans lequel se trouve son extrémité.
2. La commande de déplacement correspondante, parallèle à une bissectrice.

H2a) Algorithme de tracer dans le premier octant :

Dans cet octant, les seuls éléments d'information sont ceux correspondant à une commande de déplacement :

soit + 1 en X , 0 en Y

soit + 1 en X , + 1 en Y.



Soient A et B, les coordonnées relatives de l'extrémité du vecteur par rapport à son origine, dans le premier octant.

$$\text{Par hypothèse} \quad A \gg B \gg 0$$

Soient p et q, les coordonnées du dernier point considéré de la grille.

La distance du point (p,q) à la droite théorique est :

$$d = \frac{Bp - Aq}{\sqrt{A^2 + B^2}}$$

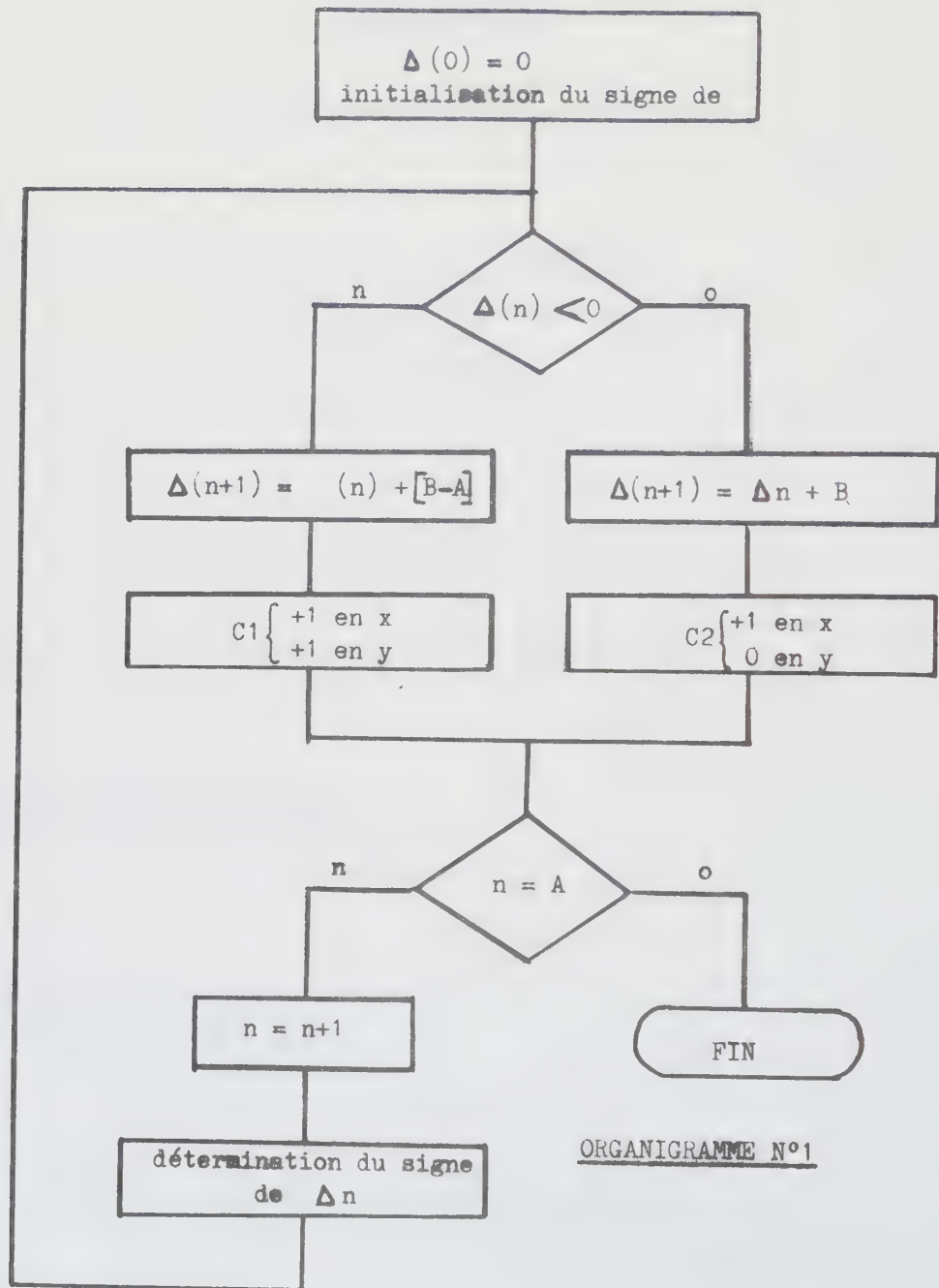
H2a1) Tracé à moins de un pas de la droite théorique

La relativité du point (p,q) par rapport à la droite théorique
dépend du signe de d :

- Si $d > 0$: le point est au dessous de la droite théorique, la commande à élaborer sera + 1 en X, + 1 en Y.
- Si $d < 0$: le point est au dessus de la droite théorique, la commande à élaborer sera + 1 en X, 0 en Y.

Le signe de d dépend du signe de $Bp - Aq = \Delta_n$ $(\sqrt{A^2 + B^2})$
étant une constante positive).

L'organigramme adopté pour le tracé d'un vecteur, dans le premier octant est le suivant :



ORGANIGRAMME N°1

Considérons le point suivant dont les coordonnées seront :

$(p + 1 ; q + 1)$ si (p, q) est au dessous de la droite théorique
 $(p + 1 ; q)$ si (p, q) est au dessus de la droite théorique.

Si $\Delta_n < 0$: $\Delta(n + 1) = B(p + 1) - Aq = \Delta_n + B$

Si $\Delta_n \geq 0$: $\Delta(n + 1) = B(p + 1) - A(q + 1) = \Delta_n + B - A$

La valeur et le signe de $\Delta(n + 1)$ sont connus si Δ_n est connu.

Le tracé étant incrémental, le point de départ, considéré comme origine, est sur la droite théorique. Donc la valeur de Δ_0 est 0.

Poser à priori le signe de Δ_0 entraînerait l'impossibilité de tracer des droites nécessitant une décomposition en plusieurs vecteurs, et parallèles aux axes ou aux bissectrices de ces axes.

Nous initialiserons donc le signe de Δ_0 à :

+ si $|A| < 2 |B|$
- si $|A| \geq 2 |B|$

H2a2) Tracé à moins de $1/2$ pas de la droite théorique :

Par hypothèse, le tracé s'effectue dans le premier octant.

Considérons le point P_n de coordonnées (p, q) . Supposons que le point P_n soit à moins de $1/2$ pas de la droite théorique.

Dans ce cas, la droite théorique coupe le segment joignant les points Q_n et R_n de coordonnées $(p+1, q-1/2)$ et $(p+1, q+3/2)$.

Pour choisir le point P_{n+1} , suivant à moins de $1/2$ pas de la droite théorique, il suffit de déterminer la relativité du point S_n de coordonnées $(p+1, q+1/2)$ par rapport à cette droite.

La relativité du point $(p+1, q+1/2)$, par rapport à la droite théorique, dépend du signe de la quantité $D_n : D_n = B(p+1) - A(q+1/2)$

Si $D_n < 0$: P_{n+1} a pour coordonnées $(p+1, q)$

Si $D_n \geq 0$: P_{n+1} a pour coordonnées $(p+1, q+1)$

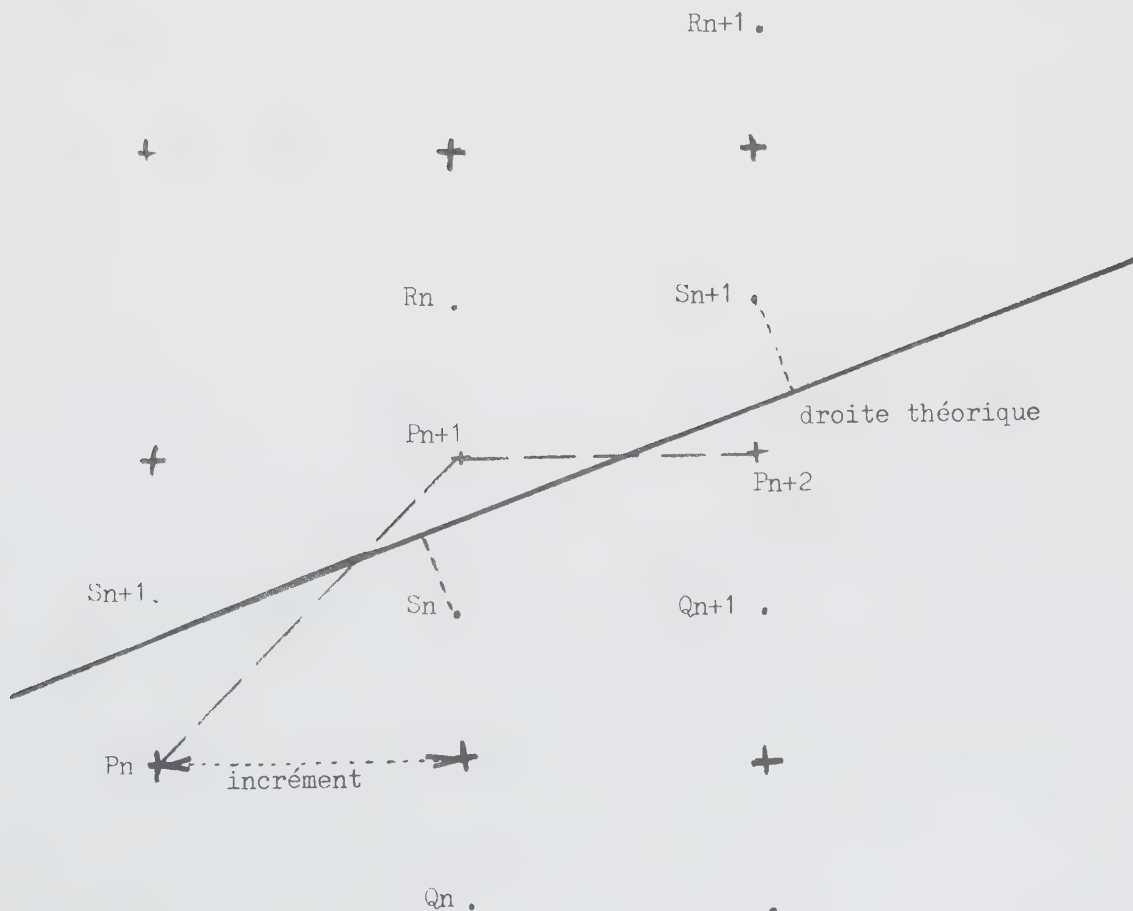


Figure H1

Le milieu du segment Q_{n+1}, R_{n+1} dépend du point P_{n+1} , donc du signe de D_n :

$$\begin{aligned} \text{Si } D_n < 0 : S_n & \text{ a pour coordonnées } (p+2, q+1/2) \\ D_{n+1} &= B(p+2) - A(q+1/2) \\ &= D_n + B \end{aligned}$$

$$\begin{aligned} \text{Si } D_n \geq 0 : S_n & \text{ a pour coordonnées } (p+2, q+3/2) \\ D_{n+1} &= B(p+2) - A(q+3/2) \\ &= D_n + B - A \end{aligned}$$

En tracé relatif, le premier point est sur la droite théorique. Sa distance à la droite théorique est donc inférieure à $1/2$ pas. Donc tous les points déterminés seront à une distance inférieure à $1/2$ pas.

$$\text{Pour } p=q=0 \quad D_0 = B - A/2$$

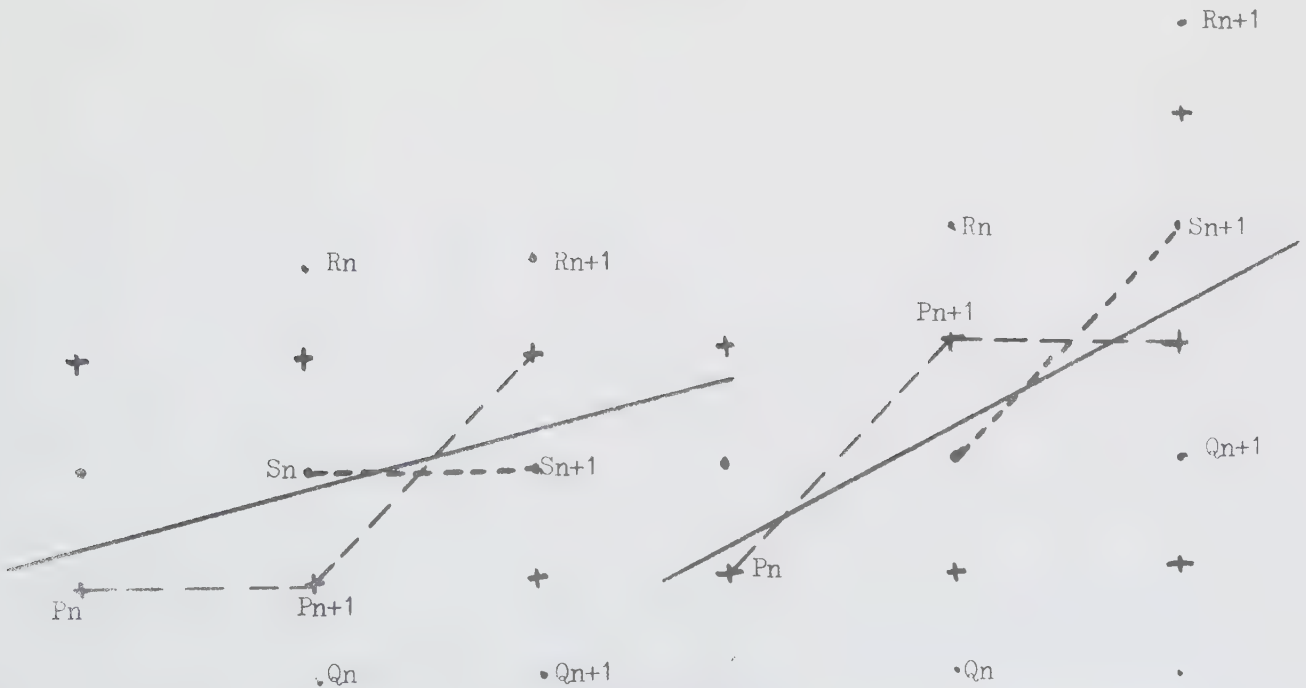


Figure H2

L'algorithme de tracé à moins de $1/2$ pas de la droite est semblable au précédent, à condition d'initialiser la valeur de Δ 0 à $B-A/2$.

H3. Définition du périphérique commandé

H3a) Table traçante :

0	rien
1	↑
2	↓
3	baisser la plume
4	→
5	↗
6	↘
7	rien
8	←
9	↖
A	↙
B	rien
C	lever la plume
D	rien
E	rien
F	rien

Le pas de la table est de 0,1mm ou 0,2mm.

H3b) Ecran

L'écran de visualisation correspondant à la table traçante est un écran à mémoire. Les commandes nécessaires sont les mêmes. Lever et baisser la plume deviennent allumer et éteindre le spot, pour l'écran. L'effacement de l'écran (F) devra être manuel.

H4. Définition de la commande de l'ULM

H4a) Commande à distance

L'ULM est commandée à distance par l'intermédiaire d'une ligne télégraphique.

Le débit de la ligne est actuellement limité à 110 bauds, par la vitesse de déplacement incrémental de la table (200 pas par seconde). Le débit pourrait être supérieur si la table travaillait à plus de 200 pas par seconde.

Les données séries sont transformées en données parallèles par un téléimprimeur, avant d'être introduites dans l'ULM.

Les caractères sont émis en code CCITT5.

Les codes 00XXXXX sont utilisés comme caractères de commandes. La longueur utile du caractère est donc limitée aux 6 bits de droite.

H4b) Commande directe

L'ULM est connectée directement à un calculateur. La longueur des éléments d'information n'a donc pas d'importance. Afin d'utiliser le même algorithme de tracé que dans le cas d'une commande à distance, nous limiterons la longueur des mots à 2 x 4 bits.

H5. Définition des éléments d'information

Chaque vecteur incrémental est généré à partir de trois éléments d'information.

H5a) Tracé à moins de 1 pas de la droite théorique

1er élément :

valeur absolue de la plus petite variation de coordonnée absolue.

2ème élément :

complément de la valeur absolue de la plus grande variation de coordonnée absolue.

3ème élément :

bit 0	<u>commande plume/spot</u>	0	baisser la plume/allumer le spot
		1	lever la plume/éteindre le spot
bit 1	<u>comparaison de $\Delta X, \Delta Y$</u>	0	$ \Delta X - \Delta Y \geq 0$
		1	$ \Delta X - \Delta Y < 0$
bit 2	<u>signe de ΔY</u>	0	$\Delta Y \geq 0$
		1	$\Delta Y < 0$

bit 3	<u>signe de X</u>	0	$\Delta X \geq 0$
		1	$\Delta X < 0$
bit 4	<u>orientation de départ du tracé</u>	0	$ A - 2 B \geq 0$
		1	$ A - 2 B < 0$
bit 5-6	11		
bit 7	<u>nature de la commande</u>	0	commande directe
	<u>de</u>	1	commande à distance

Ces trois éléments d'information permettent de tracer des vecteurs de + 64 incréments.

H5b) Tracé à moins de 1/2 pas de la droite théorique

L'initialisation à $B - A/2$ de Δ_0 nécessite :

- soit une opération de décalage à droite dans la boîte à opérations.
- soit la reception de la valeur de $A/2$.

La deuxième solution est la plus avantageuse, à condition de pouvoir recevoir le bit de droite dans un autre caractère. Il reste 2 bits libres dans le caractère contenant le numéro de l'octant et l'indication de lever ou de baisser de plume.

Le caractère A devient une grandeur codée sur 7 bits.

Le caractère B devient une grandeur codée sur 7 bits.

La transmission des caractères se fera suivant le format :

1er élément : Valeur absolue de la plus petite variation de coordonnée absolue, modulo 2^6 .

2ème élément : complément de la valeur absolue de la moitié de la plus grande variation de valeur absolue.

3ème élément :

bit 0 : lever/baisser de plume
bit 1 : bit de droite de A
bit 2 : bit de gauche de B
bit 3 : signe de ΔY , 1 si négatif
bit 4 : signe de ΔX , 1 si négatif
bit 5 : 1 si $|\Delta X| \leq |\Delta Y|$

H6. Mécanisme des échanges

H6a) Entrée d'un élément d'information

- commande à distance :

Le périphérique, le plus lent à commander, est actuellement la table traçante. Entre deux commandes, il faut attendre :

soit 35 ms s'il y a eu lever ou baisser de plume

soit 5 ms entre deux points successifs.

Lorsqu'un caractère est disponible à la sortie du receptr, la bascule BO est mise à 1. Elle sera remise à 0, par microprogramme, dès que le caractère aura été rangé dans l'ULM.

Pour une ligne télégraphique 110 **bauds**, 1 caractère est disponible environ toutes les 90 ms.

1. Le test de B0 se fera après chaque commande envoyée au périphérique.
2. Le temps maximal de tracé, lever/baisser de plume, est inférieur au temps de transmission de 3 caractères.

Donc aucun élément d'information ne peut être perdu. Il n'y a pas besoin de registres R d'entrée dans l'ULM.

Les données peuvent être prises directement à la sortie du télé-imprimeur et transférées dans les registres L.

- commande directe :

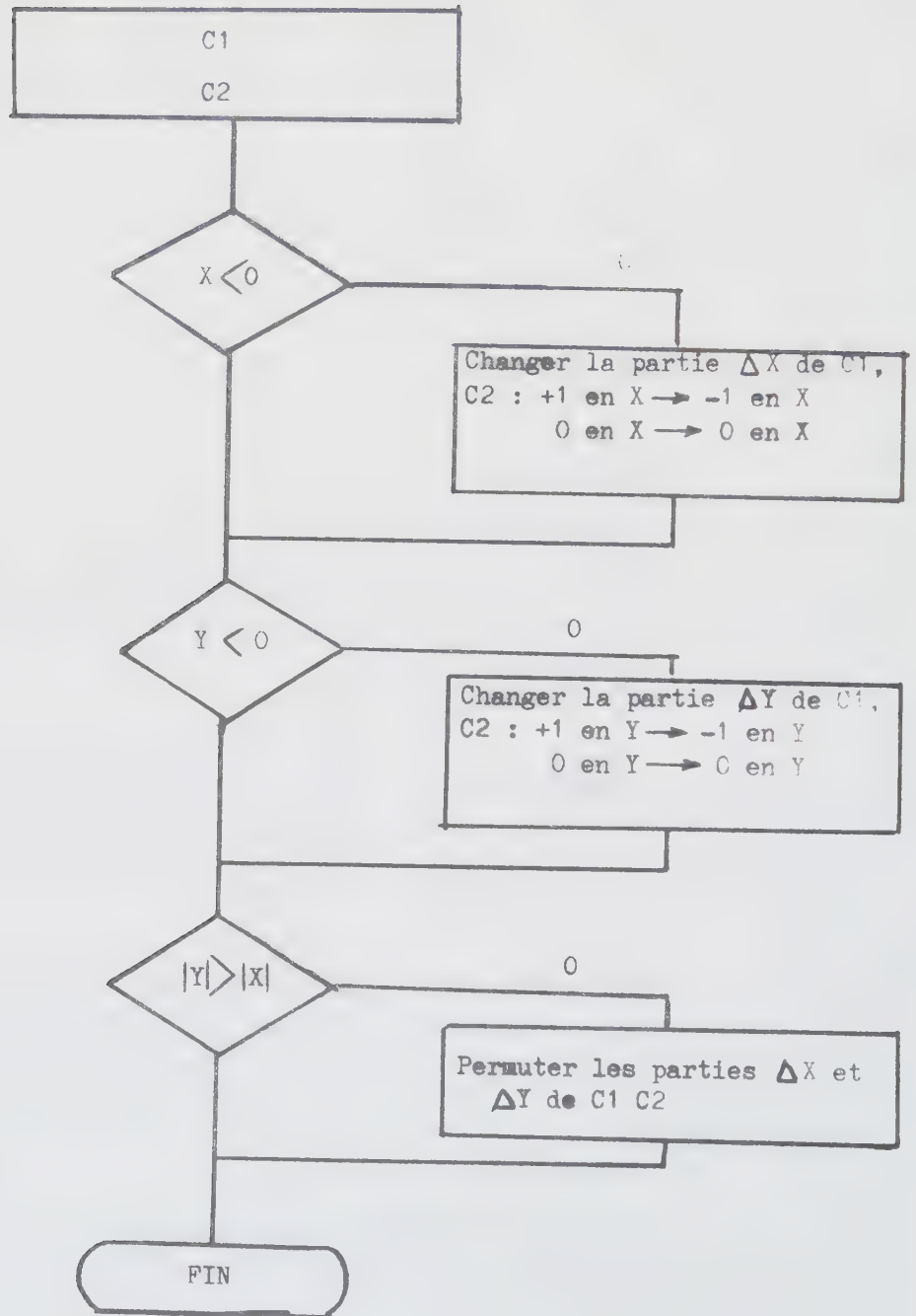
L'échange d'éléments d'information entre l'ULM et un calculateur ne peut se faire que sur l'initiative du calculateur, lorsque l'ULM est prête. Il est donc nécessaire à l'ULM de posséder deux registre réels, R, de dialogue et deux bascules réelles, B, d'état.

La bascule B0 est mise à 1, par le calculateur, au moment de l'envoi d'un élément d'information. Elle est remise à 0, par microprogramme, lorsque l'élément d'information à été transféré dans la zone traitement de l'ULM.

La bascule B1 est mise à 1, par microprogramme, lorsque les trois caractères permettant le tracé du vecteur suivant ont été transférés dans la zone traitement de l'ULM. Elle est mise à 0, par microprogramme, lorsque l'ULM est prête à recevoir les caractères nécessaires au tracé d'un nouveau vecteur.

Lorsque les bascules B0 et B1 sont à 0, le calculateur peut envoyer un élément d'information dans les registres R0 et R1 de dialogue.

ORGANIGRAMME N°2



Afin de simplifier au maximum la procédure d'échange, l'ULM enverra une interruption au calculateur, lorsque les deux bascules B0 et B1 seront à 0. L'interruption sera permanente, jusqu'à la réponse du calculateur.

H6b) Sortie d'un caractère de commande

La table traçante possède une ligne d'occupation, L'ULM élabore la commande suivante, puis vient tester la ligne d'occupation de la table. Une boucle d'attente est prévue lorsque la table n'est pas prête à recevoir.

Pour des problèmes de bruit, le signal indiquant à la table de prendre la commande suivante, doit être supérieur à 1,5 ms. Ce signal peut éventuellement durer plusieurs cycles d'ULM et un registre de dialogue entre ULM et table est nécessaire.

Si l'ULM contrôle simultanément un écran plus rapide, les commandes seront envoyées à la cadence imposée par la table.

La ligne d'occupation de la table joue le rôle d'une bascule réelle B.

H7. Ecriture du microprogramme théorique

H7a) Affectation des éléments de l'ULM :

R(0), R(1) : registres d'entrée
R(2) : registre de sortie

B(0) : 1	: un élément d'information est à transférer dans les registres L.
B(1) : 1	: les 3 caractères nécessaires au tracé du vecteur suivant ont été reçus.
B(2) : 1	: le périphérique est en train d'exécuter une commande.
L(0), L(1)	: B pour le vecteur suivant.
L(2), L(3)	: A pour le vecteur suivant.
L(5)	: partie droite du troisième caractère.
L(6), L(7)	: B du vecteur en cours de tracé.
L(8), L(9)	: B-A du vecteur en cours de tracé.
L(A), L(B)	: compteur d'incréments, initialisé à -A.
L(C), L(D)	: valeur de Δn .
L(E), L(F)	: C1, C2.
V(0) : 0	: baisser de plume/allumer le spot
: 1	: lever la plume/éteindre le spot.
V(1), V(2), V(3)	: N° d'octant.
V(4)	: bascule de travail.
V(6)	: signe de Δn .
V(8)	: bit 4 du troisième élément d'information.
V(C) : 1	: le tracé du caractère est terminé.
V(D)	: état précédent du périphérique : 0 : plume baissée 1 : plume levée.
V(E) : 1	: le premier élément d'information pour le vecteur suivant n'a pas été reçu.
V(F) : 1	: le deuxième élément d'information pour le vecteur suivant n'a pas été reçu.

H7b) Organisation du microprogramme

Le microprogramme est initialisé en 00F.

L'organigramme correspondant au microprogramme est l'organigramme 3.

Il se décompose en quatre parties :

1. Initialisation de la machine

V(C) : 1

V(D) : 1

V(E) : 1

B(0) : 0

B(1) : 0

B(2) : 1

2. Test de la bascule B0 - Rangement d'un caractère eventuel.

3. Elaboration d'une commande.

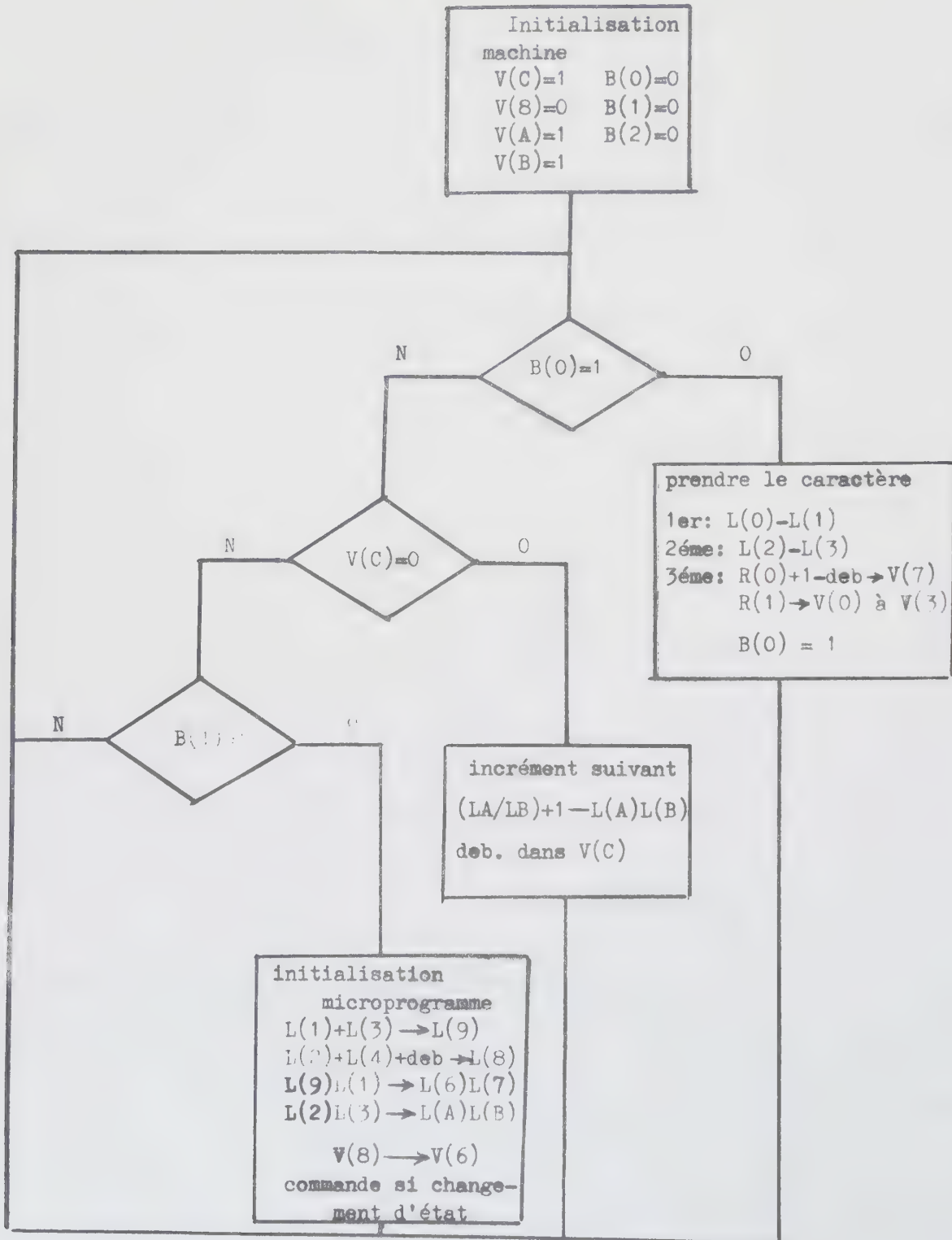
4. Initialisation du microprogramme entre deux tracés de vecteurs.

H7c) Microprogramme

Le microprogramme occupe 4 pages. Le registre M1, contenant le numéro de page, est donc un registre de 2 bits. Il sera chargé par la partie droite de l'accumulateur.

Exemple : 0011, 0111, 1011, 1111 dans A1 donnent 11 dans M1.

ORGANIGRAMME N°3



M2	M1	M0	OP	4E	4S	1E	1S	COMMENTAIRES
	0	0	K				1	
		1	KS		M1	TBO	2	
		2	S		M1	TVC		
		3		LE				
		4		LF		TV6		
		5		L7	R2			
		6		L9		TV6		
		7	A	LD				
		8	P	L6	LD		V4	
		9		L8		TV6		
		A	A	LC		RV4		
		B		LB	LC	RU	V6	
		C		LA	LB	RD		
		D			LA		VC	
		E						
		F			MO	TB2		initialisation

U L M

M2	N1	MO	OP	4E	4S	1E	1S	COMMENTAIRES
	1	0			M1	RVC		
		1	S		MO			
		2	S		M1	RU		
		3	KS		M1	TB1	1	
		4			MO	RU	V6	
		5				TVO	V4	
		6				TVD	V4	
		7				RVO		
		8	K			RU	2	
		9			R2		VD	
		A	K			TV4	B	
		B		AE	MO	RU		
		C		AE	MO	TB2		
		D			R2	TVO		
		E	P			RU		
		F		AE	LD			

U L M

M2	M1	M0	OP	AE	AS	1E	1S	COMMENTAIRES
	2	0						
		1						
		2						
		3	K	R1	M1		F	
		4		RO	L3	TVE	VF	
		5			L2	TVE	VO	
		6	S		MO	TVE	VE	
		7		RO	L1	TVF		
		8			LO	TVF	VF	
		9		RO	VS1		B1	
		A	K	LO	VS2		3	
		B			LO	TVB		
		C						
		D	S				BO	
		E						
		F						

H2	H1	H0	OP	AE	AS	1E	1G	COMMENTAIRES
	3	0	KA	AE		TV3	C	
		1			LF	RU		
		2			LC	RV8	VE	
		3	KS				5	
		4		L3	M1	RU		
		5		L2	LB	RD		
		6		L1	LA	LC	VC	
		7	A	LB	L7		B1	
		8		LO	L9			
		9	A	LA	L6	RV4		
		A	K		L8		4	
		B	K			TV1	1	
		C	KA	AE		TV2	3	
		D	KA	AE		TV3	C	
		E	K		LE		5	
		F	KA	AE		TV2	3	

H/c1) Comptage des états de chaque champ. Affectation des éléments logiques de MM :

Champ	Commande	Nombre	Élément logique	Broche de sortie
OP	K	8	8001	8001 - 08
	A	4	4001 - 1	4001 - 06
	S	5	5001 - 1	5001 - 06
	KA	4	4001 - 2	4001 - 08
	KS	3	3001 - 1	3001 - 06
	P	2	2001 - 1	2001 - 03
4E	LOI 23	5	5001 - 2	5001 - 13
	L4567	2	2001 - 2	2001 - 06
	L89AB	6	3001 - 2 - 3	3001 - 08 3001 - 12
	LCDEF	4	4002 - 1	4002 - 06
	L048C	4	4002 - 2	4002 - 08
	L159D	3	3002 - 1	3002 - 06
	L26AE	5	5002 - 1	5002 - 06
	AE	7	8002	8002 - 08
	RO	3	3002 - 1	3003 - 06
	R1	1	1001 - 1	1001 - 02
4S	LOI23	5	5002 - 2	5002 - 13
	L4567	2	2001 - 3	2001 - 08
	L89AB	6	3002 - 2 - 3	3002 - 08 3002 - 12
	LCDEF	6	3003 - 2 - 3	3003 - 08 3003 - 12
	L048C	5	5003 - 1	5003 - 06
	L159D	4	4003 - 1	4003 - 06
	L26AE	5	5003 - 2	5003 - 13
	VS12	2	2001 - 4	2001 - 11
	MO	6	3004 - 1 - 2	3004 - 06 3004 - 08
	M1	7	8003	8003 - 08
	R2	3	3004 - 3	3004 - 12

Champ	Commande	Nombre	Elément logique	Broche de sortie
1E	RV048C	5	5004 - 1	5004 - 06
	TV048C	4	4003 - 2	4003 - 08
	TV159D	2	2002 - 1	2002 - 03
	TV26AE	8	8004	8004 - 08
	TV37BF	5	5004 - 2	5004 - 13
	V0123	8	8005	8005 - 08
	V4567	6	3005 - 1 - 2	3005 - 06 3005 - 08
	V83AB	2	2002 - 2	2002 - 06
	TB0	1	1001 - 2	1001 - 04
	TB1	1	1001 - 3	1001 - 06
	TB2	2	2002 - 3	2002 - 08
	RD	2	2002 - 4	2002 - 11
	RU	8	8006	8006 - 08
1S	V048C	6	3005 - 3/006 - 1	3005 - 12 3006 - 06
	V159D-13-F-B'	6	3006 - 2 - 3	3006 - 08 3006 - 12
	V26AE-12'	6	3007 - 1 - 2	3007 - 06 3007 - 08
	V37BF-11-5'	7	8007	
	V0123-S1-1-F-C'	4	4004 - 1	4004 - 06
	V4567-1-B'	7	8008	8008 - 08
	V89AB-S2'-4-5	4	4004 - 2	4004 - 08
	B0	2	2003 - 1	2003 - 03
	B1	2	2003 - 2	2003 - 06

Les circuits nécessaires au câblage de la matrice de microprogrammation sont les suivants :

8001 à 8008	7430	8	
5001 à 5004	4931	4	
4001 à 4004	7420	4	
3001 à 3007-2	7410	7	reste 1 NAND 3
2001 à 2003-2	7400	3	reste 2 NAND 2
1001-3	7404	1	reste 3 inverseurs

L'élément logique 3007 - 3 sera utilisé pour VS12 à la place de l'élément 2001 - 4.

Les éléments logiques 2003 - 3 - 4 et 2001 - 4 seront utilisés à la place des éléments 1001 - 1 - 2 - 3.

La carte MM contiendra 26 circuits intégrés de couplage.

H7c2) Tableau des états du microprogramme

D'après le tableau précédent, il est possible de définir pour le microprogramme une correspondance entre un état et un ou plusieurs circuit de MM, pour chaque champ.

Champ	Etat	Eléments logiques	Champ	Etat	Eléments logiques
OP	K	8001			
	A	4001 - 1			
	S	5001 - 1			
	KA	4001 - 2			
	KS	3001 - 1			
	P	2001 - 1			
4E	L0	5001-2 4002-2	4S	L0	5002-2 5003-1
	L1	5001-2 3002-1		L1	5002-2 4003-1
	L2	5001-2 5002-1		L2	5002-2 5003-2
	L3	5001-2 -		L3	5002-2 -
	L4	2001-2 4002-2		L4	2001-3 5003-1
	L5	2001-2 3002-1		L5	2001-3 4003-1
	L6	2001-2 5002-1		L6	2001-3 5003-2
	L7	2001-2 -		L7	2001-3 -
	L8	3001-2-3 4002-2		L8	3002-2-3 5003-1
	L9	3001-2-3 3002-1		L9	3002-2-3 4003-1
	LA	3001-2-3 5002-1		LA	3002-2-3 5003-2
	LB	3001-2-3		LB	3002-2-3 -
	LC	4002-1		LC	3003-2-3 5003-1
	LD	4002-1		LD	3003-2-3 4003-1
	LE	4002-1		LE	3003-2-3 5003-2
	LF	4002-1		LF	3003-2-3
				MO	3004-1-2
	AE	8002		M1	8003
		3003 - 1		R2	3004-3
	RO	3003 - 1		VS1	3007-3 (2001-4) 4004-1
	R1	2001 - 4 (1001-1)		VS2	3007-3 4004-2

Champ	Etat	Eléments logiques		Champ	Etat	Eléments logiques	
1E	RV0	5004-1	8005	1S			
	RV4	5004-1	3005-1-2				
	RV8	5004-1					
	RVC	5004-1					
	RU	8006					
	RD	2002-4					
	TV0	4003-2	8005		V0	3005-3	4004-1
	TV1	2002-1	8005		V1	3006-2-3	4004-1
	TV2	8004	8005		V2		4004-1
	TV3	5004-2	8005		V3		4004-1
	TV4	4003-2	3005-1-2		V4	3005-3	8008
	TV5	2002-1	3005-1-2		V5	3006-2-3	8008
	TV6	8004	3005-1-2		V6	3007-1-2	8008
	TV7	5004-2	3005-1-2		V7	8007	8008
	TV8	4003-2	2002-2		V8	3005-3	4004-2
	TV9	2002-1	2002-2		V9	3006-2-3	4004-2
	TVA	8004	2002-2		VA	3007-1-2	4004-2
	TVB	5004-2	2002-2		VB	8007	4004-2
	TVC	4003-2			VC	3005-3	
	TVD	2002-1			VD	3006-2-3	
	TVE	8004			VE	3007-1-2	
	TVF	5004-2			VF	8007	
	TB0				B0	2003-1	
	TB1				B1	2003-2	
	TB2						

Champ	Etat	Eléments logiques	Champ	Etat	Eléments logiques
			1S (suite)	1	8007
				2	3007-1-2
				3	3006-2-3
				4	4004-2
				5	8007 4004-2
				B	3006-2-3 8008
				C	4004-1
				F	3006-2-3 4004-1

H7d) détermination du cablage de la matrice de microprogrammation :

Le cablage de la matrice de microprogrammation est obtenu en affectant à chaque microinstruction, un ensemble de broches de circuits intégrés, conformément au tableau défini au H7c2.

Le tableau ci-après définit le cablage du premier étage de la matrice de microprogrammation ne contenant que des "NAND" à 2 , 3, 4, 5, 8 entrées.

Adresse	OP	4E		4S		1E		1S	
00	8001-01	-	-	-	-	-	-	8007-01	-
01	3001-03	-	-	8003-01	-	2003-09	-	3007-03	-
02	5001-01	-	-	8003-02	-	4003-09	-	-	-
03	-	4002-01	5002-01	-	-	-	-	-	-
04	-	4002-02	-	-	-	8004-01	3005-03	-	-
05	-	2001-04	-	3004-01	-	-	-	-	-
06	-	3001-09	3002-05	-	-	8004-02	3005-04	-	-
07	4001-01	4002-04	3002-03	-	-	-	3005-09	-	-
08	2001-02	2001-05	5002-02	3003-09	4003-01	-	-	3005-01	8008-01
09	-	3001-10	4002-09	-	-	8004-03	3005-05	-	-
0A	4001-02	4002-05	4002-10	-	-	5004-01	3005-09	-	-
0B	-	3001-11	-	3003-10	5003-01	8006-01	-	3007-04	8008-02
0C	-	3001-01	5002-03	3002-09	-	2002-12	-	-	-
0D	-	-	-	3002-10	5003-09	-	-	3005-02	-
0E	-	-	-	-	-	-	-	-	-
0F	-	-	-	3004-03	-	2002-09	-	-	-
10	-	-	-	8003-03	-	5004-02	-	-	-
11	5001-02	-	-	3004-04	-	-	-	-	-
12	5001-03	-	-	8003-04	-	8006-02	-	-	-
13	3001-04	-	-	8003-05	-	2003-12	-	8007-02	-
14	-	-	-	3004-05	-	8006-03	-	3007-05	8008-03
15	-	-	-	-	-	4003-10	8005-11	3005-13	8008-04
16	-	-	-	-	-	2002-01	-	3006-03	8008-05
17	-	-	-	-	-	5004-03	8005-12	-	-
18	8001-02	-	-	-	-	8006-04	-	3007-09	-
19	-	-	-	3004-02	-	-	-	3006-09	-

Adresse	OP	4E	4S	1E	1S
1A	8001-03	- -	- -	4003-12 3005-10	3006-10 8008-06
1B	-	8002-01 -	3004-09 -	8006-05 -	- -
1C	-	8002-02 -	3004-10 -	2002-10 -	- -
1D	-	- -	3004-13 -	4003-13 8005-01	- -
1E	2001-01	- -	- -	8006-06 -	- -
1F	-	8002-03 -	3003-11 4003-02	- -	- -
20	-	- -	- -	- -	- -
21	-	- -	- -	- -	- -
22	-	- -	- -	- -	- -
23	8001-04	2001-12 -	8003-06 -	- -	3006-11 4004-10
24	-	3003-03 -	5002-09 -	8004-04 -	8007-03 -
25	-	- -	5002-10 5003-10	8004-05 -	- -
26	5001-04	- -	3004-11 -	8004-06 -	3007-10 -
27	-	3003-04 -	5002-11 4003-04	5004-09 -	2003-02 -
28	-	- -	5002-12 5003-02	5004-10 -	8007-04 -
29	-	3003-05 -	3007-01 4004-01	- -	2003-04 -
2A	8001-05	5001-09 002-12	3007-02 4004-09	- -	3006-01 -
2B	-	- -	5002-08 5003-03	5004-11 2002-04	- -
2C	-	- -	- -	- -	- -
2D	5001-05	- -	- -	- -	2003-01 -
2E	-	- -	- -	- -	- -
2F	-	- -	- -	- -	- -
30	4001-09	8002-04 -	- -	5004-12 8005-02	- 4004-02
31	-	- -	3003-01 -	8006-11 -	-
32	-	- -	3003-02 5003-04	5004-04 2002-05	3007-11
33	3001-05	- -	- -	- -	8007-05 4004-04
34	-	5001-10 -	8003-11 -	8006-12 -	- -

Adresse	OP	4E	4S	1E	1S
35		5001-11 5002-04	3002-11		
36		5001-12 3002-04	3002-01 5003-11		3006-04
37	4001-04	3001-13	2001-09		2003-05
38		5001-08 4002-13	3002-02 4003-05		3006-05 8008-11
39	4001-05	3001-02 5002-05	2001-10 5003-12	5004-05 3005-11	
3A	8001-06		3002-13 5003-05		4004-12
3B	8001-11			2002-02 8005-04	8007-06
3C	4001-10	8002-05		8004-11 8005-05	3006-02
3D	4001-12	8002-06		5004-08 8005-03	4004-05
3E	8001-12		3003-13 5003-08		8007-11 4004-13
3F	4001-13	8002-11		8004-12 8005-06	3006-13

Lorsqu'un état nécessite l'utilisation de deux circuits, les sorties correspondantes servent d'entrée à un deuxième étage ne contenant que des "NOR" (ou "OR"). Le cablage de cet étage est donné dans un tableau supplémentaire : connections entre éléments de la matrice.

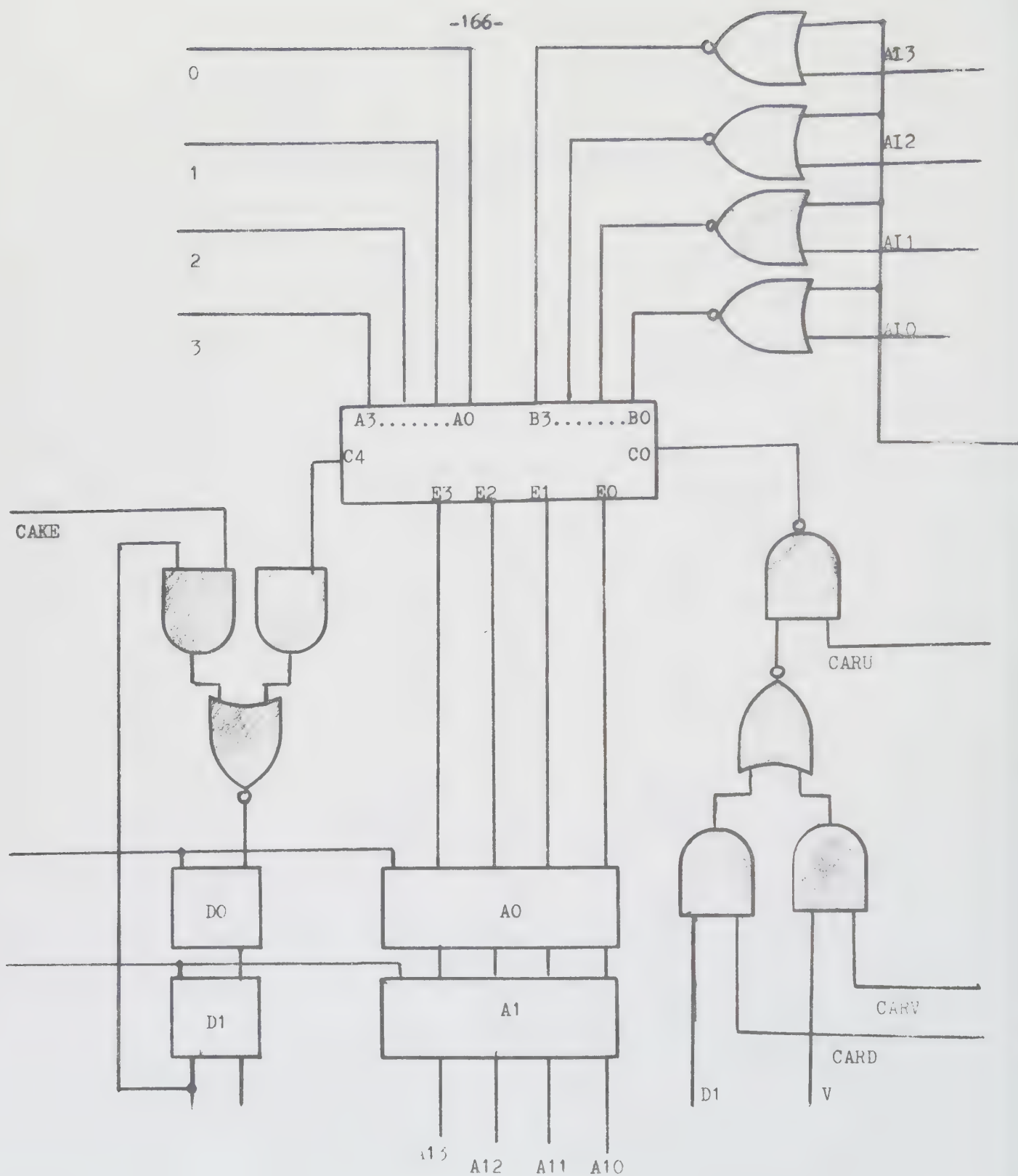
Le tableau suivant donnera les liaisons entre la sortie des états de la matrice de microprogrammation.

Connections entre éléments de la matrice :

Liaisons :

Sortie :

3001-08	2801-03	
3001-12	2801-02	2801-01
3002-08	2801-06	
3002-12	2801-05	2801-04
3003-08	2801-08	
3003-12	2801-09	2801-10
3004-06	2801-11	
3004-08	2801-12	2801-13
3005-06	2802-03	
3005-08	2802-02	2802-01
3005-12	2802-03	
3006-06	2802-05	2802-04
3006-08	2802-08	
3006-12	2802-09	2802-10
3007-06	2802-11	
3007-08	2802-12	2802-13



H8. Choix des composants de l'ULM

Pour réaliser l'ULM, nous avons choisi la série TTL 7400. Outre son faible prix et sa rapidité d'approvisionnement, cette série présente quelques circuits particulièrement adaptés à la structure de l'ULM. En particulier un additionneur parallèle 4 bits (7483) et un ensemble de 4 x 4 latch (74170).

H8a). Opérateur traitement (fig. H1)

L'examen du microprogramme montre en champ OP l'absence de certaines commandes :

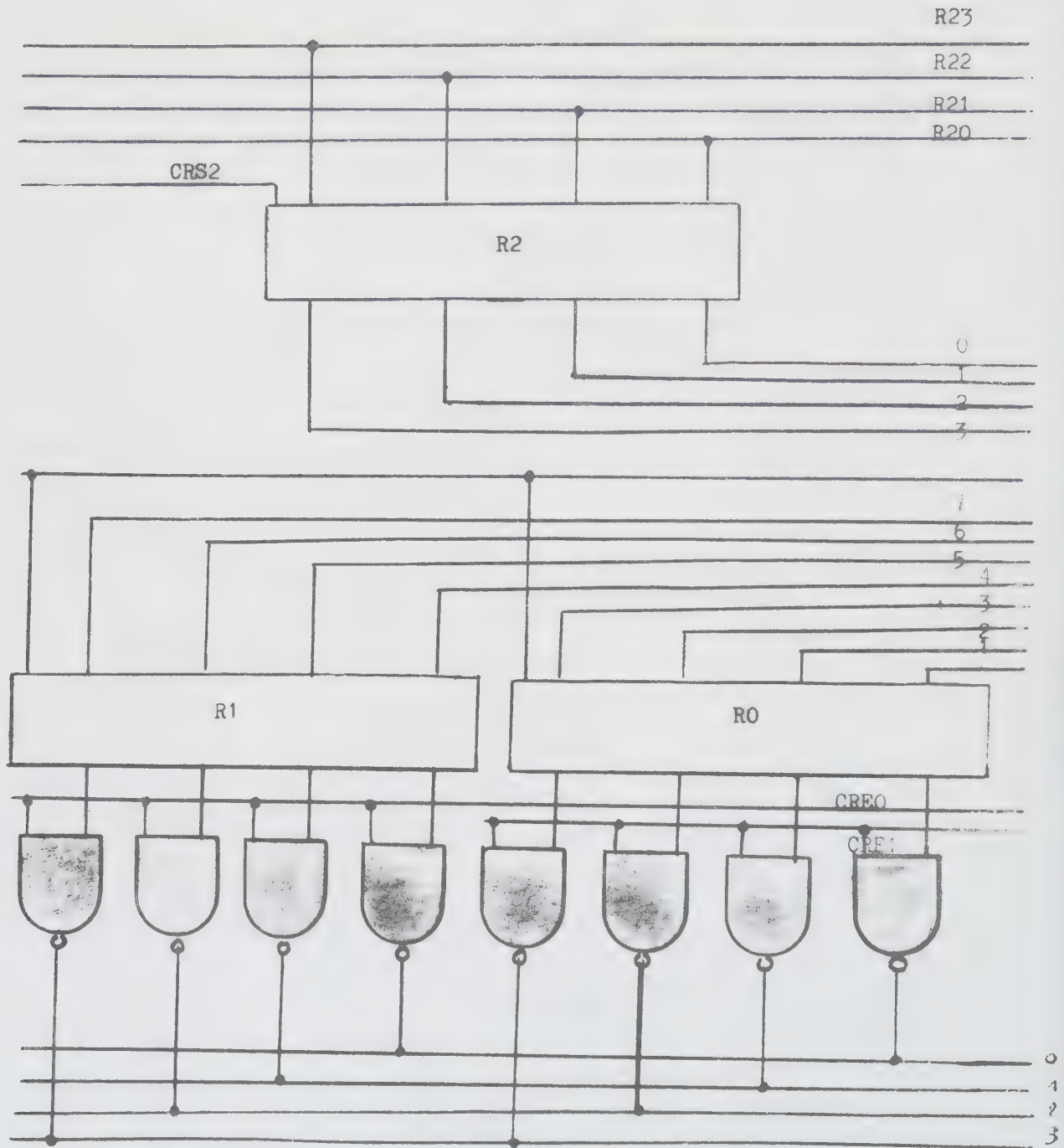
- complémentation
- opération spéciale réalisée à la recopie de AO dans A1.

La boîte à opérations comprend :

1 additionneur parallèle 4 bits	7483 - 1
2 registres AO et A1	7475 - 2
2 bascules DO et D1	7475 - 1
1 barrière de 1er opérande	7402 - 1
1 ensemble logique pour le report initial	7451 - 1/2
	7400 - 1/4
1 ensemble logique pour le débordement	7451 - 1/2

H8b) 1. Opérateur L

Le traitement nécessite l'emploi de 14 registres. Il est plus simple et moins coûteux de prendre, dans ce cas, un opérateur L complet de 16 registres.



H2, ENSEMBLE R

L'opérateur L comprend :

16 registres de 4 bits	74170 - 4
1 barrière d'écriture	7400 - 1

2. Ensemble R (fig. H2)

L'ensemble R comprend

2 registres d'entrée R0 et R1	7475 - 2
2 barrière d'accès dans la BOP	7403 - 2
1 registre de sortie R2	7475 - 1

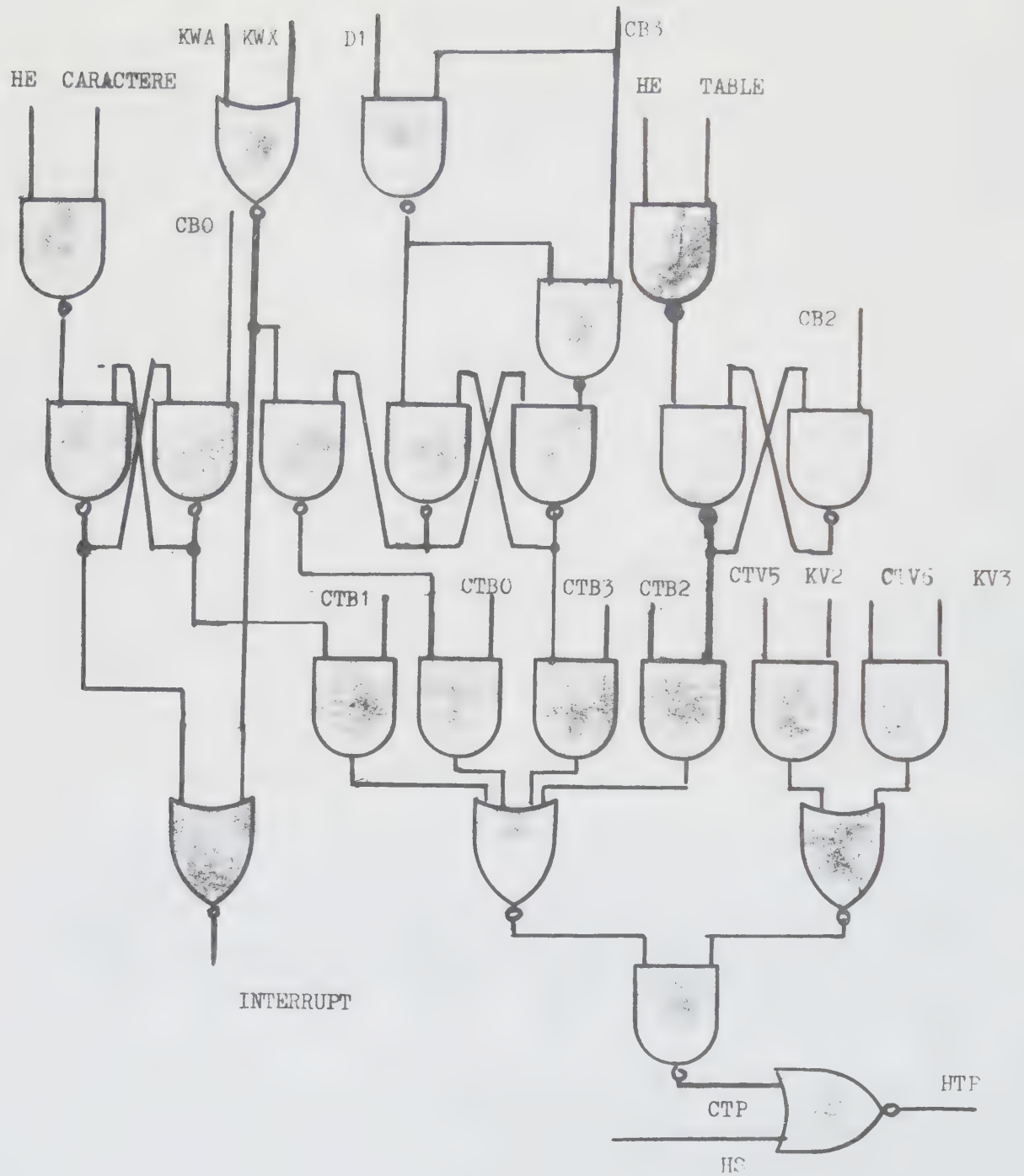
H8c) 1. Opérateur V

Le traitement nécessite l'emploi de 11 bascules V. Nous prendrons un opérateur complet 16 x 1 bit.

L'opérateur V est constitué d'un circuit 4 x 4 "latches", utilisé pour L.

La lecture d'une bascule se fait par adressage, en lecture, d'un ensemble de 4 bits, et choix à la sortie d'un bit parmi quatre, à l'aide d'un circuit 4 ET/OU/INVERSE.

L'écriture dans une bascule se fait par adressage, en lecture et en écriture, du registre contenant cette bascule. L'entrée et la sortie sont bouclées par l'intermédiaire d'un 2 ET/OU/INVERSE. Le choix de la bascule à modifier se fait à l'entrée. Les 3 autres bascules se recopient dans elles-mêmes.



H3. Ensemble B

Le microprogramme permet la lecture d'une bascule V(RV, TV) et l'écriture dans la même bascule ou dans une autre, au cours de la même microinstruction. Les opérations de lecture et d'écriture seront donc successives (voir §C). Dans ce cas, il est nécessaire de mémoriser la valeur de la bascule lue.

Remarque : L'addition de 4 2ET/OU/INVERSE et de 4 NOR à 2 entrées permet d'utiliser le circuit 74170, tantôt en 16 x 1 bits, tantôt en 4 x 4 bits, avec la possibilité d'utiliser les deux arrangements, au cours d'une même microinstruction.

L'opérateur V comprend :

- 1 Ensemble 4 x 4 bits : 74170 - 1
- 1 Ensemble de choix d'un bit parmi 4 (lecture) :
7454 - 1
- 1 Ensemble de choix d'un bit parmi 2 (écriture):
7451 - 2
- 1 Ensemble d'adressage lecture : 7451 - 1
- 1 Ensemble de choix en arrangement 16X1 ou 4X4 :
7451 - 2
- 1 bascule de sortie : 7475 - 1/4

2.Ensemble B (fig.H 3)

L'ensemble B comprend :

- 1 bascule BO d'entrée, "latch set-reset", mise à 1 par le monde extérieur et à 0 par le microprogramme. 7400 - 1/2

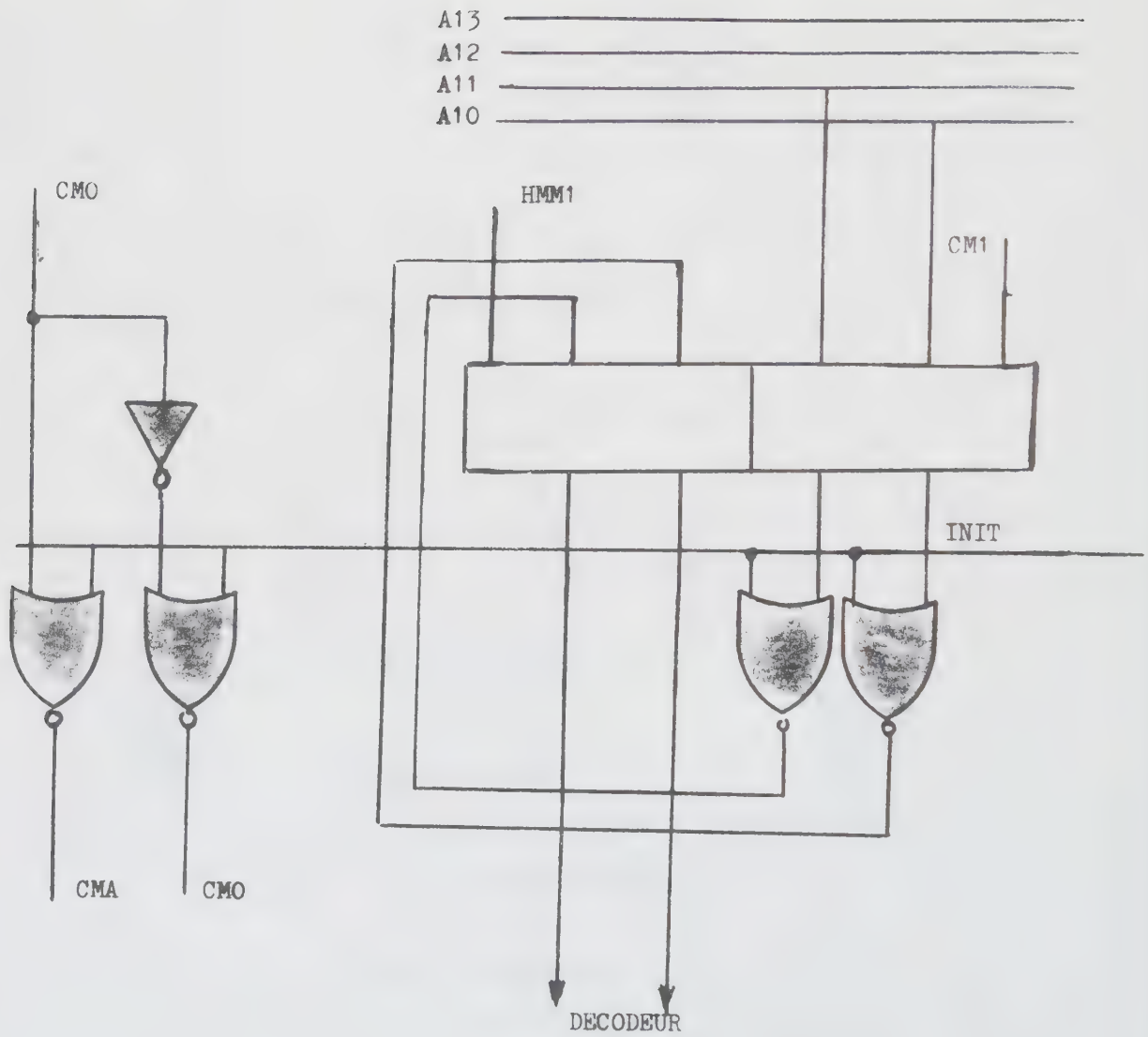


Fig H4. REGISTRES MO, MM1 - COMMANDE DE MO-

1 bascule B1 d'indication des 3 caractères acquis
pour le tracé suivant. 7475 - 1/4

La bascule B2 représente le bit d'état de la table
Elle n'appartient pas à la structure, mais peut être testée par le microprogramme.

H8d) Matrice de microprogrammation

1. Opérateur MA (Fig. E3b et H4)

Le microprogramme est divisé en 4 pages de 16
lignes. Le registre M1 est un registre de 2 bits, chargé avec la partie droite
de l'accumulateur.

L'opérateur MA comprend :

Les registres d'adressage M0 MM0, M1 MM1 : 7475 - 3

Un ensemble logique effectuant l'opération + 1 :

7486 - 3/4

7402 - 2/4

7404 - 1/6

Un ensemble logique de choix entre MM0+1 et A :

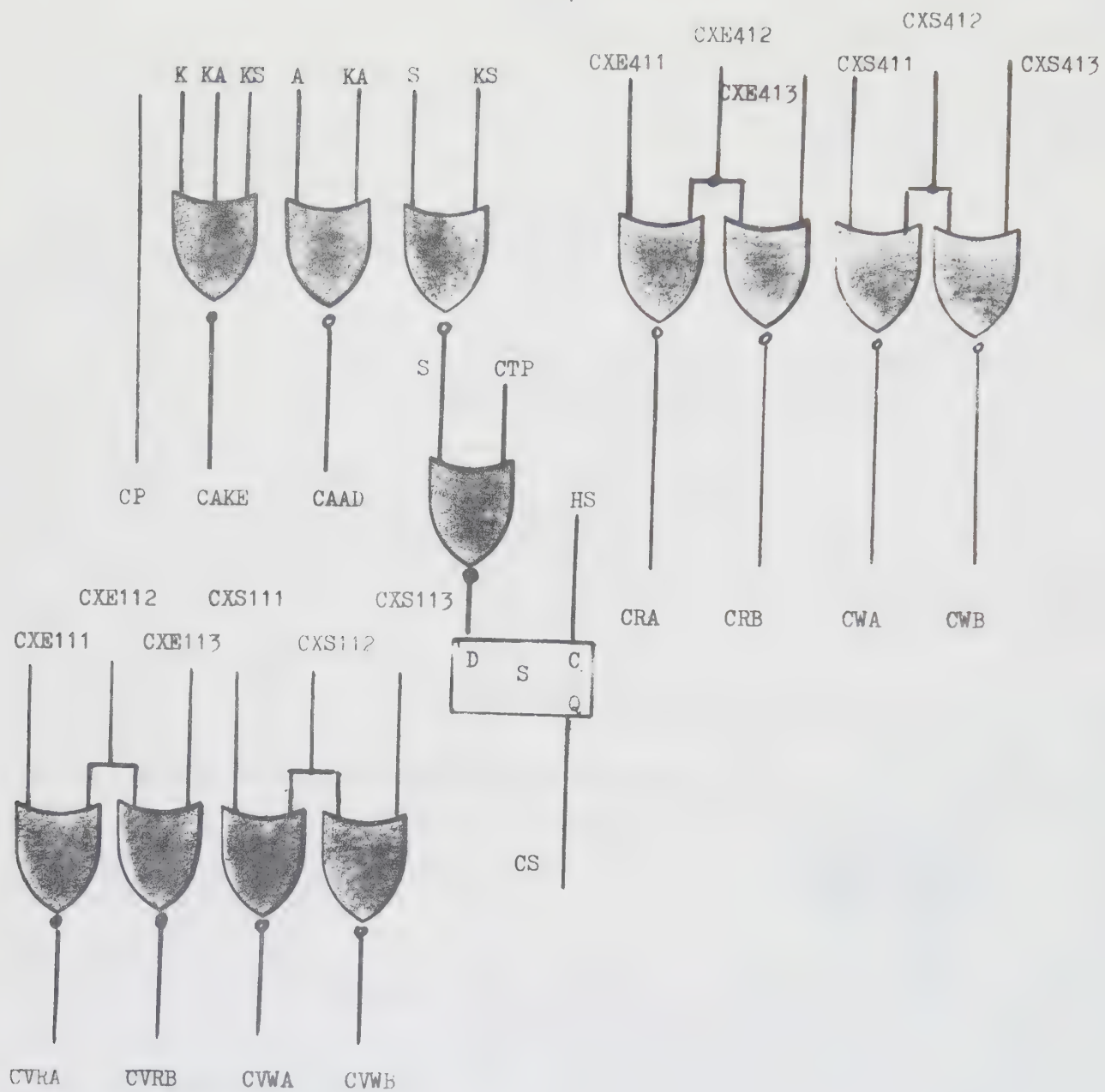
7451 - 2

Un ensemble d'initialisation à l'adresse 00F :

7402 - 1

2. Ensemble MM

Le câblage de l'ensemble MM dépend du microprogramme
c'est à dire du nombre de microinstructions et du nombre d'utilisation d'un



même état.

L'ensemble MM est constitué de NAND à 2, 3, 4, 5, 8 entrées et de NOR à 2 entrées.

Les notations utilisées pour les circuits sont celles définies au paragraphe E2a).

3. Ensemble MS (fig.H5)

Champ OP

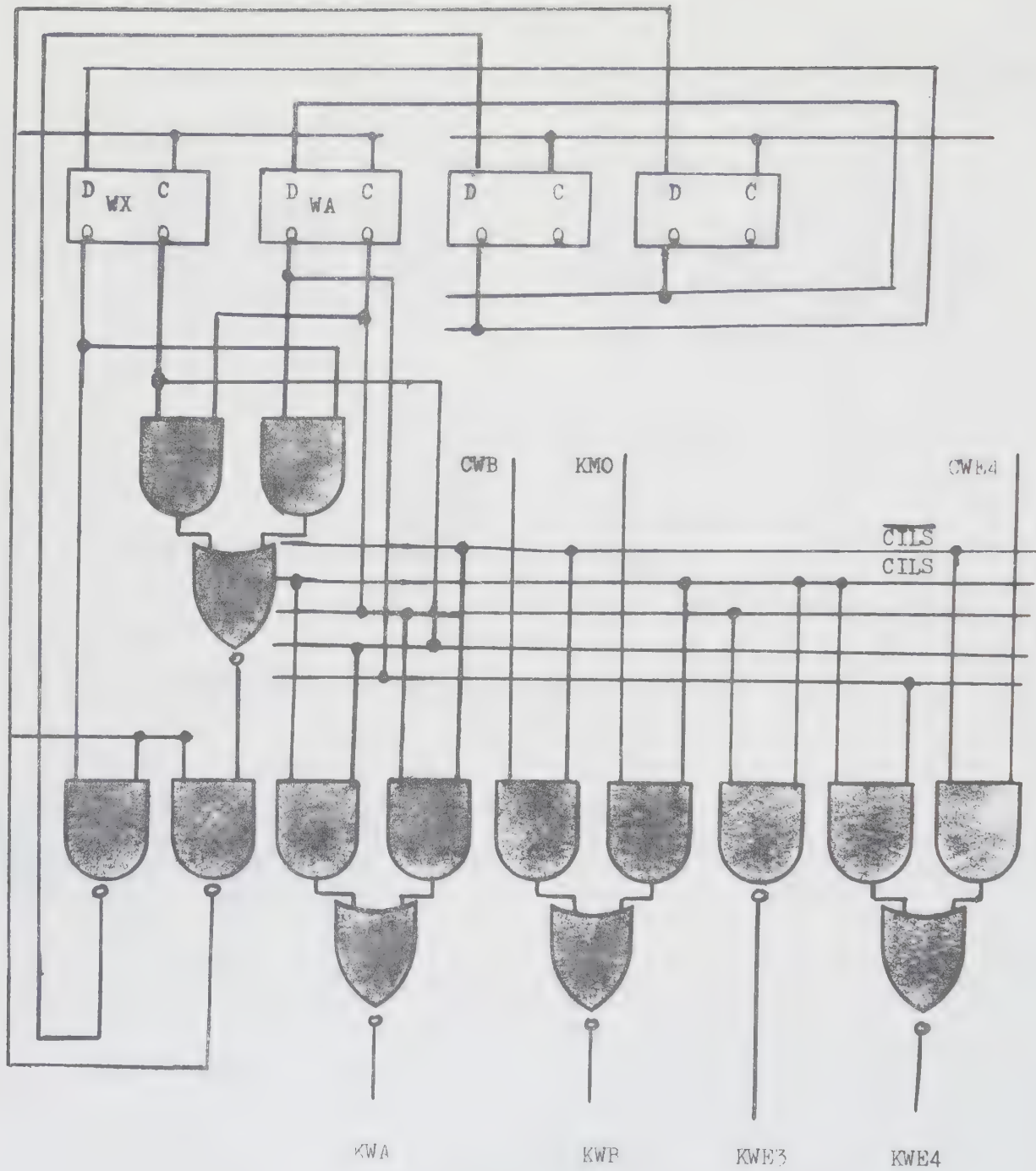
Les commandes élaborées par l'ensemble MM sont :

K - A - S - KA - KS - P.

L'ensemble MS doit élaborer les commandes :

K - A - S - P.

La commande S doit être mémorisée pour le cycle suivant, si la microinstruction adressée est exécutable.



Gestion de R4

H9. Etude du microprogramme

H9a) Etude générale

Le microprogramme, du paragraphe H7, utilise les opérateurs V et L, en adressage direct, par la seule matrice de microprogrammation.

Associations à L deux registres réels R4 et R5, pour l'adresser en lecture et en écriture.

1. Acquisition de 3 caractères

L'acquisition des 3 caractères nécessaires au tracé du vecteur suivant nécessite :

- 10 microinstructions dont :
 - 4 pour ranger le premier caractère
 - 10 pour ranger le second et le troisième caractère
- 2 bascules, VE et VF, avec initialisation de VE à 1 .
- 1 bascule, B1.

En utilisant le registre R4, l'acquisition nécessite :

- 2 microinstructions de rangement de R0 et R1.
- 4 microinstructions de gestion de R4.

2. Choix conditionnel entre deux registres L (Fig. H6a)

Le choix conditionnel entre deux registres L est équivalent à un adressage lecture par un registre R5 dont 3 bits sont déterminés et dont le quatrième dépend de la valeur de la bascule de condition.

L'utilisation de R5 permet de gagner 3 microinstructions soit 12 éléments de couplage.

H9b) Etude en fonction de l'application "Générateur de vecteurs"

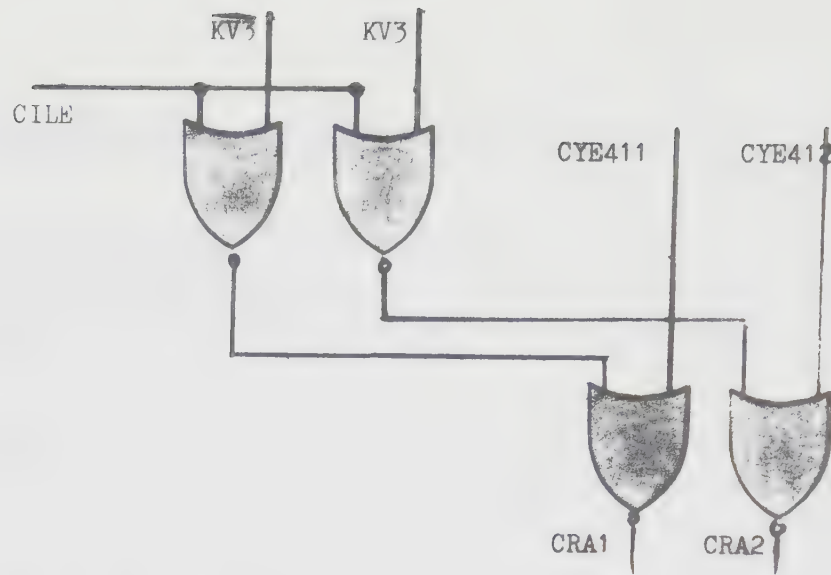
1. Gestion de R4

Les trois caractères sont rangés dans des registres L consécutifs, au cours de microinstructions consécutives. Le bit de droite de R4 est identique au bit de droite de M0.

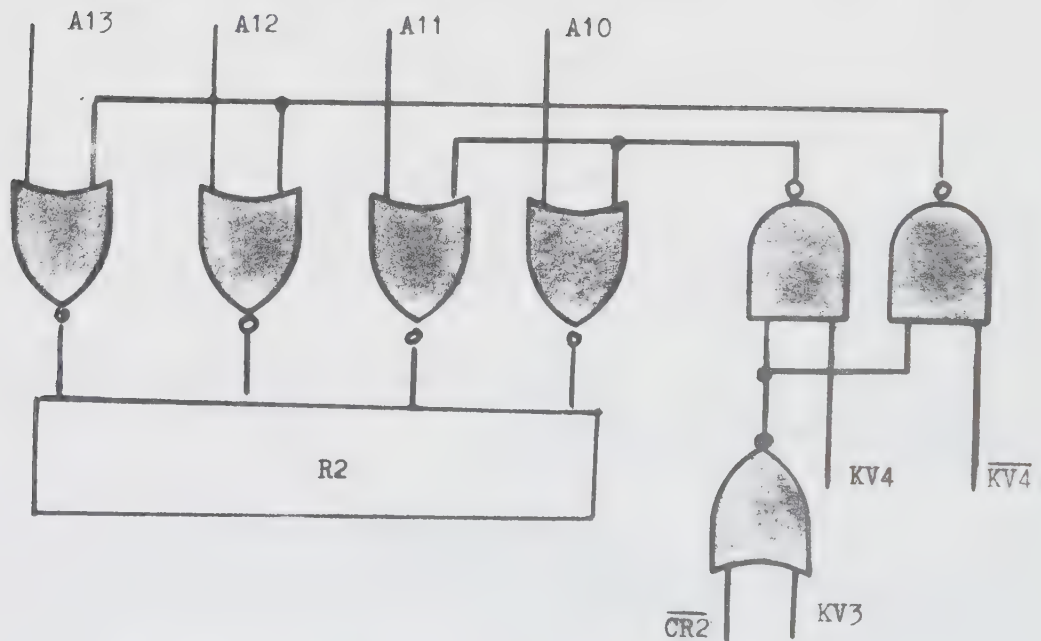
Trois caractères correspondent à six registres L. Le bit de gauche de R4 peut être fixe.

La gestion de R4 se ramène donc à la gestion d'un registre de 2 bits sur lequel il faut faire +1 à chaque caractère rangé.

Cette opération est très simple à réaliser en Hardware et permet d'économiser les 4 microinstructions de gestion de R4.



H6a. Gestion de R5



H6b. Ecriture dans R2

Comparaison des deux solutions :

Gain solution 2

8 microinstructions

47 éléments de couplage

2 bascules V

1 bascule B

Gain solution 1

1 commande H pour réaliser +1

1 élément de couplage

1 Quad-latch
3 AND/OR/INVERT 2 entrées
2 NAND 2 entrées

2. Gestion de R5

Pour choisir entre les constantes B et B-A, il suffit d'adresser B et de modifier éventuellement le dernier bit de R5 suivant la valeur de V6, bascule de la structure.

Pour choisir entre C1 et C2, il suffit d'adresser C1 et de modifier le dernier bit de R5 suivant la valeur de V6.

Les trois premiers bits de R5 étant fixes, leur existence hardware n'est pas nécessaire. Il n'y aura aucune microinstruction de gestion de R5.

Comparaison des deux solutions :

Gain solution 2

3 microinstructions

12 éléments de couplage

Gain solution 1

3. Ecriture dans R2 (Fig. H6b)

La commande 02 est identique à la commande C1, masquée à droite ou à gauche, suivant l'octant de tracé.

La commande lever/baisser de plume correspond à la constante F, masquée à droite ou à gauche suivant la nature de la commande.

L'écriture dans R2 peut donc se faire à travers un étage d'éléments logiques, masquant ou non la partie droite ou la partie gauche de l'ensemble de 4 bits à écrire.

Comparaison entre les deux solutions :

<u>Gain solution 2</u>	<u>Gain solution 1</u>
6 microinstructions	-
30 éléments de couplage	-
	6 NOR à 2 entrées

4. Opération OU exclusif du 2 bascules V

Les bascules V sont disponibles 4 par 4 à la sortie de l'opérateur V. Il est donc très simple de réaliser un OU exclusif sur 2 bascules appartenant à un même ensemble de 4.

Comparaison des deux solutions :

<u>Gain solution 2</u>	<u>Gain solution 1</u>
3 microinstructions	-
11 éléments de couplage	-
1 bascule V	1 OU exclusif

H9c) Conclusions

Le microprogramme de la solution 1 comporte , relativement à la solution 2 :

- 57 microinstructions
- 215 éléments de couplage
- 3 bascules V (VE, VF, V4)
- 1 bascule B (B1)

Le microprogramme de la solution 2 comportera :

- $57-20 = 37$ microinstructions, soit un gain de 35 %
- $215-100 = 115$ éléments de couplage, soit un gain de 47 %
- 1 Quad latch
- 2 NAND 2 entrées
- 3 AND/OR/INVERT 2 entrées
- 6 NOR 2 entrées.

La solution 1 est relative à un microprogramme de tracé à moins de 1 pas de la droite théorique.

Le gain obtenu par passage de la solution 1 à la solution 2 permet d'envisager la microprogrammation du tracé à moins de 1/2 pas de la droite théorique qui avait été abandonné dans la solution 1, car le microprogramme dépassait 4 pages et nécessitait un adressage de la matrice de microprogrammation plus complexe.

H10. Microprogrammation réelle

H10a) Affectation des éléments de l'ULM

R(0), R(1)	: registres d'entrée
R(2)	: registre de sortie
B(0)	: un élément d'information est à transférer dans les registres L
B(1)	: tracé terminé - 3 caractères reçus
B(2)	: état du périphérique
L(0)	: partie droite de B - vecteur en cours de tracé
L(1)	: partie droite de $\bar{A}$ - " " "
L(2)	: partie gauche de $\bar{A}$ - " " "
L(3)	: partie droite de B-A
L(4)	: partie gauche de B - " " "
L(5)-L(6)	: valeur de Dn
L(7)	: partie gauche de B-A
L(8)-L(9)	: B pour le vecteur suivant
L(A)-L(B)	: $\bar{A}/2$ " "
L(C)-L(D)	: N° d'octant - nature du tracé - informations complémentaires sur A et B
L(F)	: commande
V(0)	: Lever/baisser pour le tracé suivant

- V(1) : 7ème bit de $\bar{A}$
- V(3) : état du périphérique (lever/baisser)
- V(4) : 7ème bit de B
- V(5) : signe de ΔY pour le tracé suivant
- V(6) : signe de ΔX pour le tracé suivant
- V(7) : 1 si $|\Delta X| < |\Delta Y|$ pour le tracé suivant
- V(A) : signe de D_n
- V(B) : 1 si $|\Delta X| < |\Delta Y|$

H10b) Organisation du microprogramme

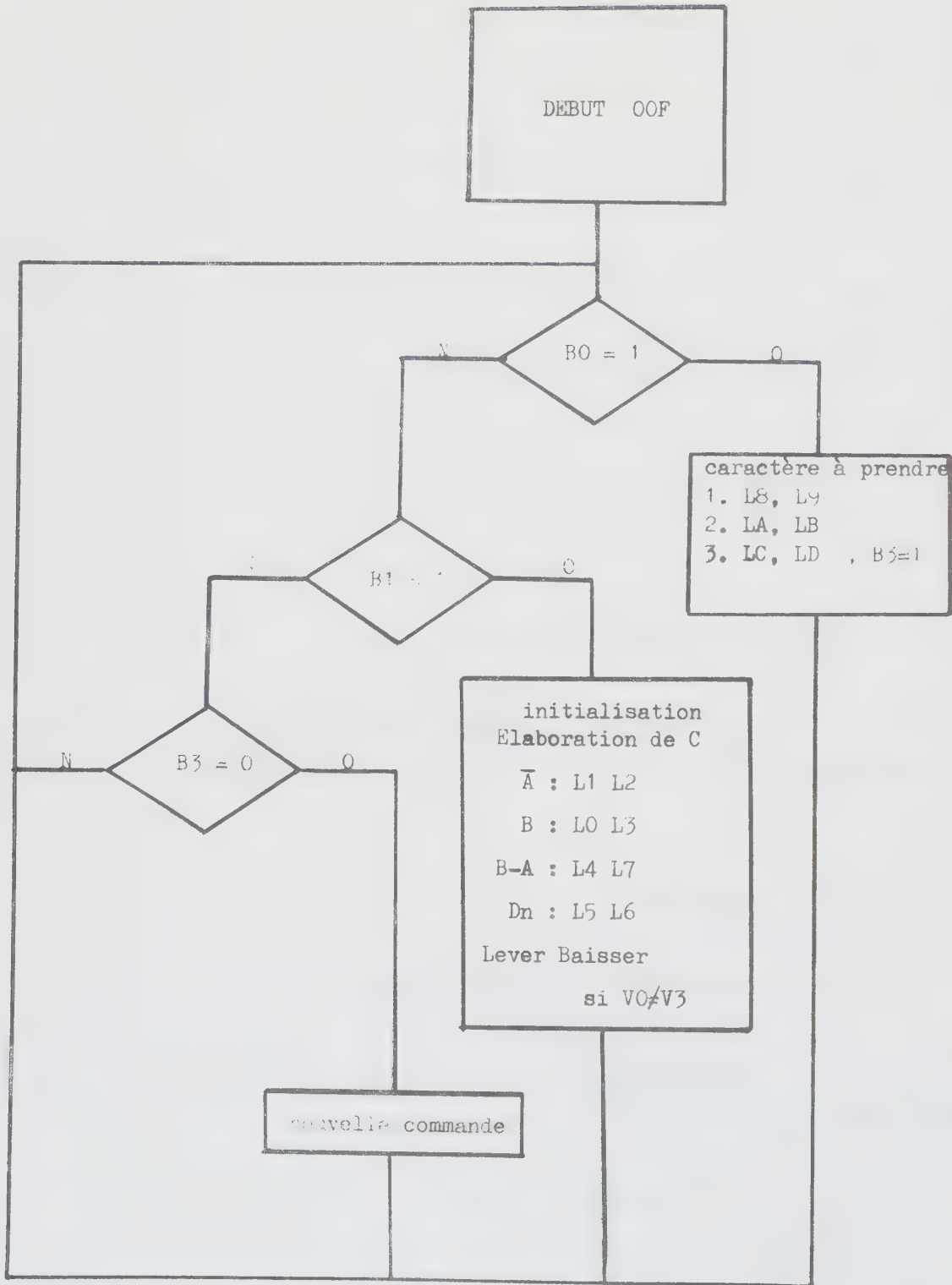
L'organisation du microprogramme est identique à celle définie au paragraphe H7. Il n'y a pas d'initialisation des bascules.

H10c) Microprogramme

Le microprogramme occupe 3 pages.

Le registre M1 est un registre de 2 bits, chargé à partir de la partie droite de l'accumulateur.

Le microprogramme correspond à l'ordinogramme suivant :



U L M

H2	M1	N0	0P	4E	AS	1E	1S	COMMENTAIRES
0	0	0	K				5	
		1	K		MO	TBO	1	
		2	H(1)	R1				Adressage écriture L : 9 B D 9.. 8 A C 8..
		3		RO	IL			L9 - LB - LD
		4	K		IL	RU	0	L8 - LA - LC
		5	KS		M1	TB1	5	→ 2 - 6
		6	K		MO	TB3	5	
		7		LF				
		8		IL	R2			L0 - L4 suivant VA
		9	A	L5				
		A	KP	IL	L5		F	L3 - L7 suivant VA
		B	A	L6		RD		
		C		L1	L6	RU	VA	
		D		L2	L1	RD		
		E			L2		B3	
		F		AE	MO	TB2		INITIALISATION

M2	M1	M0	OP	4E	4G	1E	1S	COMMENTAIRES
	2	0		LB	L4		V1	
		1	A	LB		RV5		
		2	A	L3	L2	RV1	B3	
		3			L7	RV3		V3 = 1 Si VO ≠ V3 V3 = 0 Si VO = V3
		4		AE	M1			
		5	S	AE	MO			→ 3-0 Si V3 = 1 → 0-F Si V3 = 0
		6	KA	AE		TV6	3	← 0-5
		7	KA	AE		TV5	C	
		8			LF	TV7		
		9		L8			VB	
		A	A	LA	L0			
		B	K	L9	L5		3	
		C	A	LB	L3	RD		
		D		LA	L6	RU	VA	
		E	A	LA		RV1		
		F	A	L8	L1		V5	

-187-

U L M

N2	N1	N0	OP	4E	4S	1E	1S	COMMENTAIRES
	3	0		AE	MO	TB2		← 2-5
		1			R2			
		2	P		M1	RVO		Lever ou baisser suivant V0
		3					V3	
		4	S		MO			→ 0-F
		5						
		6						
		7						
		8						
		9						
		A						
		B						
		C						
		D						
		E						
		F						

H10d) Etude du microprogramme

1. L'étude du microprogramme montre qu'il n'existe aucune microinstruction contenant à la fois une lecture et une écriture, dans l'opérateur V.

L'adressage en lecture de V pourra être simplifié, les bascules servant à mémoriser le report ou le test ne sont plus nécessaires :

Gain : 1 Quad Latch

2. Considérons les champs 4S et 1S : il y a 4 microinstructions dans lesquelles il existe simultanément une écriture dans L et V

H2	H1	H0	OP	4E	4S	1E	1S	COMMENTAIRES
0	0	C		L1	L6	RU	VA	
0	2	O		LB	L4		V1	
0	2	D		LA	L6	RU	VA	
0	2	F		L8	L1		V5	

Nous pouvons remarquer que la partie gauche des champs 4S et 1S est identique, pour ces 4 microinstructions. Il existe d'ailleurs, une correspondance entre L et V, qui est respectée dans ce microprogramme :

L 048C	V 0123
L 159D	V 4567
L 26AE	V 89AB
L 37BF	V CDEF

Il existe 3 microinstructions de plus, contenant L en champs 4S et pour lesquelles le champ 1S contient une constante.

M2	M1	M0	OP	4S	4S	1S	1S	COMMENTAIRES
0	0	4	K	IL	RU	0		
0	0		KP	IL	L5	F		
0	2		K	L9	L5	3		

Par analogie avec la correspondance précédente, nous avons entre L et la constante :

L 0123	2 6 AE
L 4567	3 7 BF
L 89AB	0 4 8C
L CDEF	1 5 9D

Cette correspondance est aussi respectée dans le microprogramme. Il est donc possible de rendre commune la partie gauche de 4S et la partie gauche de 1S :

Gain : 7 éléments de couplage

3. Considérons les microinstructions qui modifient

l'opérateur V :

M2	M1	M0	OP	A5	A5	1E	1S	COMMENTAIRES
0	0	3		RO	IL			
0	0	4	K		IL	RU	O	
0	0	C		L1	L6	RU	VA	
0	2	0		LB	L4		V1	
0	2	9		L8			VB	
0	2	D		LA	L6	RU	VA	
0	2	F	A	L8	L1		V5	
0	3	3					V3	

La bascule de débordement D1 est copiée dans :

V1 - V3 - V5 - VA - VB.

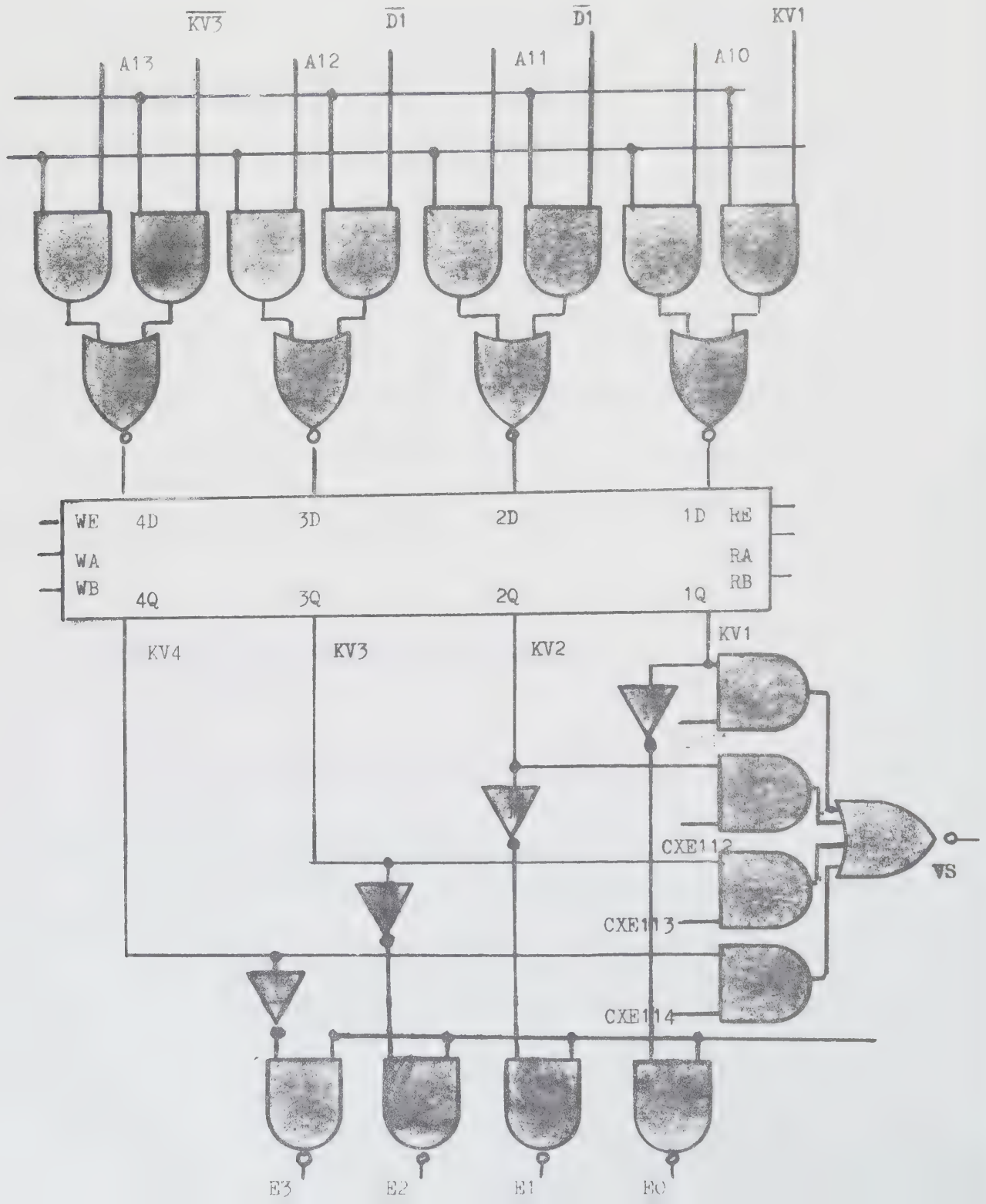
V3 et VB sont des doublures de V0 et V7.

La recopie de V0 dans V3 et de V7 dans VB se fait

par l'initialisation de D1/A1 à 0/1111. Le résultat est :

soit 0/1111 si V0 ou V7 = 0

soit 1/0000 si V0 ou V7 = 1



La recopie se fait durant l'initialisation pour un nouveau tracé.

Pendant l'initialisation, les valeurs des trois autres bascules V associées à V3 et VB n'ont aucune signification. Il est donc possible d'écrire, dans l'ensemble des 4 bascules, non le débordement, mais la valeur de A1.

La valeur des bascules V0, V3, V4, V7, V8, VB doit être conservée pendant tout le tracé.

La valeur des bascules V1 - V5 - VA est modifiable.

Il est donc possible d'économiser un étage de AND/ OR/ INVERT, à l'entrée de l'opérateur V en effectuant le câblage suivant :

Les bascules:	V0	-	-	V3
	V4	-	-	V7
	V8	-	-	VB
	VC	-	-	VF

sont bouclées sur elles-mêmes et sont accessibles par le bus 4S

Les bascules:	-	V1	V2	-
	-	V5	V6	-
	-	V9	VA	-
	-	VD	VE	-

sont accessibles par le bus 1S et par le bus 4S.

Sans aucune contrainte, l'opérateur V peut donc être simplifié et passer de 7 circuits intégrés à 4.

4. Remarque

Les modifications apportées au Hardware nécessitent une étude approfondie du microprogramme, du cablage de la matrice de microprogrammation et des opérateurs.

Dans le cas du générateur de vecteurs, cette étude est simple et rapide.

Pour une application plus complexe, l'étude des modifications pourra être longue et il faut trouver un compromis entre, d'une part le temps d'étude nécessaire, et d'autre part l'économie apportée, en fonction du nombre d'ULM à réaliser pour une même application. Cette étude est rentable pour des applications simples construites en grande série.

H10e) Cablage de la matrice de microprogrammation

Champ	Etat	Nombre	circuits logiques	circuit de sortie
OP	K	6	8001	2801-1
	KA	2	8001 - 2001-1	
	KS	1	1003-2 - 3001-1	
	A	8	8002	2801-2
	S	2	3001-1	
	P	2		
4E	L0123	3	3001-2	2801-3
	L4567	3	3001-3	
	L89AB	5	3002-2 - 2001-4	
	LCDEF	1	1002-6	

Champ	Etat	Nombre	circuits logiques	circuit de sortie
4E (suite)	L048C	3	3002-1	2801-4
	L159D	2	2001-3	
	L26AE	9	8003 1001-3	
	AE	6	8004	
	RO	1	1001-1	
	R1	1	1001-2	
4S	L0123	6	4001-1 2002-2	2802-3
	L4567	5	4002-1 2002-3	2802-4
	L89AB	0		
	LCDEF	3	3005-2	
	L048C	5	3003-3 2002-1	2801-1
	L159D	3	3004-1	
	L26AE	6	3004-2 3004-3	2801-2
	IL	2	2004-1	
	R2	2	2003-4	
	MO	6	3003-1 3003-2	
	R1	3	3002-3	
1E	RV	6	8005	2803-1 2803-2
	V0123	5	3006-2 2002-4	
	V4567	5	3005-1 2004-3	
	V89AB	2	2003-1	

Champ	Etat	Nombre	circuits logiques	circuit de sortie
1E (suite)	V048C	1	1001-6	
	V159D	3	3006-3	
	V26AE	1	1001-5	
	V57BF	1	1001-4	
	TV	2	1002-1 1002-2	
	B0	1	1002-3	
	B1	1	1002-4	
	B2	2	2003-2	
	B3	1	1002-5	
1S	V	4	4001-2	
	B2	2	2004-2	
	B3	2	2003-3	
	'O''A''3'	4	4002-2	
	'C''F'	2	2004-4	

Implantation des circuits logiques

00	8001								
01	8001	3003-2	1002-3	4002-2					
02		1001-2	1001-1						
03		2004-1	1001-1						
04	8001	2004-1	3005-3	1003-1	4002-2				
05	3001-1	1003-2	3002-3	1002-4					
06	8001	3003-1	1002-5						
07		1002-6							
08		3001-3	2003-4	2003-1					
09	8002	3001-2	2001-3						
0A	8001	2001-2	3003-3	4002-1	2003-1		2004-4		
0B	8002	3002-2	2001-3	3006-1	2004-2				
0C	4001-2	2002-3	3001-2	3002-1	3004-3		3005-3		
0D	3002-2	3002-1	2002-2	2002-1	3006-1				
0E	3004-2	2002-3	2003-3						
0F	8004	3003-2	2003-2						
20	1001-3	3004-1	4002-1	4001-2					
21	8002	8003	3006-2	3006-3	8005				
22	8002	2004-3	3006-3	3002-1	3004-3	2002-2	8005	2003-3	
23	8005	2004-3	2002-3	1001-5					
24	8004	3002-3							
25	8004	3001-1	3003-2						
26	8004	8001	2001-1	3003-3	2002-4		1002-2	4002-2	
27	8004	8001	2001-1	3004-2	2002-4		1002-1	2004-4	
28	3005-2	8005	3006-2	1001-4					

29	3001-3	8003	3004-2	3005-2			
2A	8002	2001-4	8003	3004-1	4001-1		
2B	8001	3001-2	8003	2002-1	4002-1	3006-2	4002-2
2C	8002	8003	3006-1	4001-1			
2D	4002-1	3002-2	8003	3004-3	4001-2	3005-3	
2E	8002	2001-4	8003	8005	3005-1	3006-3	
2F	8002	3001-3	8003	3003-3	4001-1	4001-2	
30	8004	3003-1	2003-2				
31	2003-4		3005-1				
32	2001-2	3002-3	8005	1001-6	3005-1		
33	3004-1	3005-2	2004-2				
34	3001-1	3003-2					

Liaisons entre les circuits intégrés

00	8001-01						
01	8001-02	3003-03	1002-05	4002-09			
02		1001-03					
03		1001-01	2004-01				
04	8001-03		2004-02	3005-01	1003-01	4002-10	
05	3001-03	1003-03	3002-01	1002-09			
06	8001-04	3003-04	1002-11				
07			1002-13				
08		3001-01	2003-12	2003-01			

09	8002-01 3001-09 2001-09	
0A	8001-05 2001-05 3003-01 4002-02 2003-02	2004-12
0B	8002-02 3002-09 2001-10 3006-03 2004-04	
0C	4001-09 2002-09 3001-10 3002-04 3004-01	3005-02
0D	3002-10 3002-03 2002-04 2002-01	3006-04
0E	3004-09 4001-01 2003-09	
0F	8004-01 3003-09 2003-04	
20	1001-05 3004-04 4002-04	4001-10
21	8002-03 8003-01 3006-09 3006-02	8005-04
22	8002-04 2004-09 3006-01 3004-02 3002-05 2002-05 8005-05 2003-10	
23	8005-01 2004-10 2002-01 1001-11	
24	8004-02 3002-02	
25	8004-03 3001-05 3003-11	
26	8001-06 8004-04 2001-01 3003-02 2002-12	1002-03 4002-12
27	8001-11 8004-05 2001-02 3004-10 2002-13	1002-01 2004-13
28	3005-09 8005-02 3006-10	1001-09
29	3001-02 8003-02 3004-11	3005-11
2A	8002-05 2001-12 8003-03 3004-05	4001-02
2B	8001-12 3001-11 8003-04 2002-02 3006-11 4002-05	
2C	8002-06 8003-05 3006-05	4002-13
2D	4002-01 3002-11 3005-13 8003-06 3004-13	4001-04
2E	8002-11 2001-13 3005-05 8003-11 8005-06	4001-12
2F	8002-12 3001-13 4001-05 8003-12 3003-13	3006-13
		4001-13
30	8004-06 3003-05 2003-05	
31	2003-13 3005-03	
32	2001-04 3002-13 8005-03 1001-13	3005-04
33	3004-03 3005-10 2004-05	
34	3001-04 3003-10	

Connections entre éléments de la matrice de microprogrammation

Liaisons	Sortie	Commande
8001-08 2801-02		
1003-04 2801-03	2801-01	<u>CAKE</u>
2801-01 1003-05	1003-06	CAKE
8002-08 2801-05		
2001-03 2801-06	2801-04	<u>CAAD</u>
	3001-06	CS
	2001-06	CP
	3001-08	CXE411
	3001-12	CXE412
3002-08 2801-08		
2001-11 2801-09	2801-10	<u>CXE413</u>
	1001-02	CRO
	1001-04	CR1
	3002-06	CYE411
	2001-08	CYE412
8003-08 2801-11		
1001-06 2801-12	2801-13	<u>CYE413</u>
	1002-12	CYE414
	8004-08	CAAE
3003-12 2802-02		
2002-03 2802-03	2802-01	<u>CXS411</u>
	3004-06	CXS412
3004-08 2802-05		
3004-02 2802-06	2802-04	<u>CXS413</u>
	2004-03	
	3002-12	

Liaisons	Sortie	Commande
4001-06 2802-08		
2002-06 2802-09	2802-10	<u>CYS411</u>
4002-06 2802-11		
2002-08 2802-12	2802-13	<u>CYS412</u>
	3005-08	CYS413
3003-06 2803-08	2003-11	CR2
3003-08 2803-09	2803-10	<u>CMO</u>
3006-08 2802-02		
2004-08 2803-06	2803-01	<u>CXE111</u>
	2803-04	<u>CXE112</u>
	2003-03	CXE113
	1002-06	CTB0
	1002-08	CTB1
	2003-06	CTB2
	1002-10	CTB3
	1001-08	C4E111
	1001-10	C4E112
	3006-12	C4E113
	1001-12	C4E114
	1002-02	CTV5
	1002-04	CTV6
	8005-08	CARV
	3005-12	CARU
	3006-06	CARD
	4001-08	CXS111
	2004-11	CXS112 (4K)

Liaison	Sortie	Commande
	4002-08	CXS113 (3K)
	2003-08	CYS111 (B3)
	2004-06	CYS112 (B2)
	1003-10	CYS113 (B0)

H11. Comparaison des deux solutions

Le tableau suivant résume les différences de conception et de hardware des deux solutions.

	élément de comparaison	SOLUTION 1	SOLUTION 2
MM	Tracé	Moins 1 pas de la droite théorique	moins de 1/2 pas de la droite théorique
	Microinstruction	57	37
	Elément de couplage	215	152
Opérateurs	Hardware	74170-1 7451-1	7400-1/2 7402-3/2

Conclusions :

La solution 2, plus précise, nécessite moins d'éléments de couplage pour la matrice de microprogrammation.

Au point de vue hardware, le cablage et la gestion du registre d'adressage écriture de L est compensé par la simplification de l'opérateur V. L'économie de la solution 2 par rapport à la solution 1 est celle d'un circuit 74170, dont le prix est environ 25 fois celui des 7400, 7402 et 7451.

LA SOLUTION 2 EST RETENUE POUR LA REALISATION D'UN GENERATEUR DE VECTEURS A COMMANDE DIRECTE OU A COMMANDE PAR TELEIMPRIMEUR.

H12. Implantation de l'ULM

1. Implantation des éléments

Les cartes choisies, pour réaliser l'ULM, peuvent comporter jusqu'à 15 circuits intégrés 14 ou 16 broches et sont munies d'un connecteur de 64 broches.

Pour respecter la structure modulaire de l'ULM, nous pouvons cabler un opérateur ou un ensemble par plaque.

Cependant le nombre de circuits utilisés, d'une part pour L et R, d'autre part pour B et V, montre qu'il est souhaitable de regrouper sur une même carte l'Opérateur Stockage et l'Ensemble correspondant.

L'ULM comportera cinq parties :

Carte OP	Boîte à opérations
Carte LR	Opérateur L - Ensemble R
Carte VB	Opérateur V - Ensemble B
Carte MA	Opérateur MA
Carte MS	Ensemble MS

La matrice de microprogrammation sera cablée sur une carte spéciale MM, pouvant contenir 35 circuits intégrés.

2. Implantation de l'unité

Pour chaque ULM, l'implantation peut être différente, suivant la nature de l'application et son utilisation.

Le but du chapitre H était de montrer comment peut s'étudier et se réaliser une ULM. Un plan détaillé de cablage est donc nécessaire après chaque étude, mais ne peut être donné ici qu'à titre indicatif.

Pour réaliser un prototype, nous choisirons les éléments standards définis au paragraphe W. Le nombre de circuits intégrés étant limité, inférieur à 80, il est évident qu'une application industrielle entraînerait une implantation différente, par exemple sur une ou deux cartes, au maximum

Les paragraphes suivants se rapporteront donc à l'Unité réalisée au laboratoire.

H12a) Matrice de microprogrammation

La matrice de microprogrammation, avec son ensemble de décodage est implantée sur une carte standard pouvant recevoir 35 circuits intégrés 14 broches.

Le cablage de la carte est donné au paragraphe H10e).

L'implantation du connecteur correspondant est la suivante :

CARTE .		MM					
	A		COMMENTAIRE		A		COMMENTAIRE
A1		GND		B1		GND	
A2	1003-	10	CYS 113	B2			
A3				B3	2803-	10	CMO
A4	2803-	11	CXE 111	B4	2003-	11	CR2
A5	2803-	04	CXE 112	B5	3005-	08	CYS 413
A6	2803-	03	CXE 113	B6	2802-	13	CYS 412
A7	1002-	06	CTB 0	B7	2802-	10	CYS 411
A8	1002-	08	CTB 1	B8	3002-	12	CMI
A9	2003-	06	CTB 2	B9	2004-	03	GILS
A10	1002-	10	CTB 3	B10	4002-	08	CXS 113
A11	1001-	08	CYE 111	B11	2004-	11	CXS 112
A12	1001-	10	CYE 112	B12	2802-	04	CXS 413
A13	3006-	12	CYE 113	B13	3004-	06	CXS 412
A14	1001-	12	CYE 114	B14	2802-	01	CXS 411
A15	1002-	02	CTV 5	B15			
A16	1002-	04	CTV 6	B16	8004-	08	CAAE
A17	8005-	08	CARV	B17	1002-	12	CYE 414
A18	3005-	12	CARV	B18	2801-	13	CYE 413
A19	3006-	06	CARD	B19	2001-	08	CYE 412
A20	4001-	08	CXS 111	B20	3002-	06	CYE 411
A21	2003-	08	CYS 111	B21	1001-	04	CRI
A22	2004-	06	CYS 112	B22	1001-	02	CRO
A23		D		B23	2801-	10	CXE 413
A24		D		B24	3001-	12	CXE 412
A25		C		B25	3001-	08	CXE 411
A26		B		B26	2001-	06	CP
A27		A	Entrée de MM	B27	3001-	06	CS
A28		Q2		B28	2801-	04	CMAD
A29		Q2		B29	1003-	06	CAKE
A30		Q1		B30	2801-	01	CAKE
A31		Q1		B31			

H12b) Autres cartes

Le câblage des autres cartes correspond aux schémas donnés au cours du paragraphe H.

L'implantation du connecteur, étant conforme au paragraphe E, elle se déduit de ces schémas.

I. CONCLUSION

L'étude des Unités Logiques Modulaires Microprogrammées est le résultat du travail d'une équipe. Les divers aspects de l'ULM ont été étudiés parallèlement.

- Définition.
- Etude et réalisation hardware des Opérateurs et Ensembles.
- Etudes de diverses applications
 - Gestion de machine à écrire, par ULM ayant un accès direct à la mémoire du calculateur principal.
 - Gestion de périphériques travaillant à distance.
 - ULM à Ensemble de Dialogue effectuant, sous contrôle du microprogramme, la conversion Analogique - Digitale pour n voies ($n=16$ à 256)
- Réalisation d'une ULM "Générateur de vecteurs."
- Ecriture d'un simulateur modulaire et "d'outils software" destinés à la mise au point et à la réalisation rapides d'Unités spécialisées.

Un grand nombre de modifications, depuis l'étude du projet initial, ont été apportées au fur et à mesure de l'avancement des travaux de chaque groupe. Notre expérience dans le domaine des ULM permet, maintenant, de publier la synthèse des travaux effectués. Le présent document fait partie de l'ensemble des publications ULM.

B I B L I O G R A P H I E

- B. Etude et réalisation d'opérateurs permettant par leur assemblage la constitution de structures microprogrammées générales ou spécialisées - Définition -

M. FRADIN Thèse 3ème cycle 1971

- C. ELBIT 100 Reference Manual.

- D. "ORBIT". Cours sur les circuits intégrés.

- E. Catalogue de divers fabricants de circuits intégrés.

- F.-Simulateur ULM. Projet Programmeur d'Etudes 70.71

- Définition du langage PATOIS d'utilisation en temps réel du simulateur ULM. Travail P.E 70.71

-Etude de la détermination automatique du cablage de la matrice de microprogrammation d'une ULM. Travail P.E 70.71

DOCTORAT :
THESE DE : DOCTEUR INGENIEUR

NOM : BOSSUET Gérard

Prénoms : Gérard

né le 13/5/44 . . 1944. à . GENOZAC Dép^t : Charente Maritime

1ère THESE : ETUDE ET REALISATION D'OPERATEURS PERMETTANT PAR LEUR .

ASSEMBLAGE LA CONSTRUCTION DE STRUCTURES MICROPROGRAMMEES GENERALES OU
SPECIALISEES . - ETUDE PRATIQUE

2e THESE (Propositions faites par l'Université) :



Vu et Approuvé

PARIS, le 27 septembre 1971

Le Président

de l'Université de PARIS VI

A handwritten signature in dark ink, appearing to be "J. J. J." or similar, written over a light background.

